



PHP 程式設計基礎

1-1 註解 comment

當我們作任何計畫，當參與的人一多，每個人的想法或作法都不同，在程式中，最簡單就是分成模組來寫，把模組組成一個大的計畫，而每個模組的程式都要有註解(comment)，這樣其它開發或維持的人才看的懂。註解就是表示在程式沒有作用的。

//這符號後面的文字是當作註解。

/*.....*/這符號裏面所包的就是註解。

#這個符號就像 shell 的註解。

範例 comment.php

只有第九行會顯示出資料，其它第二行到第七行為註解，在程式中是沒有作用的。

```
1 <?php
2     echo "測試"; // 這就像是c語言的註解,只作用在這一行
3     /* 這個
4     註解
5     是可
6     以多
7     行*/
8
9     echo "One Final Test"; #這就像是shell里面的註解
10 ?>
```

顯示出資料”測試 one final test”，其它第二行到第七行為註解，在程式中是沒有作用的。

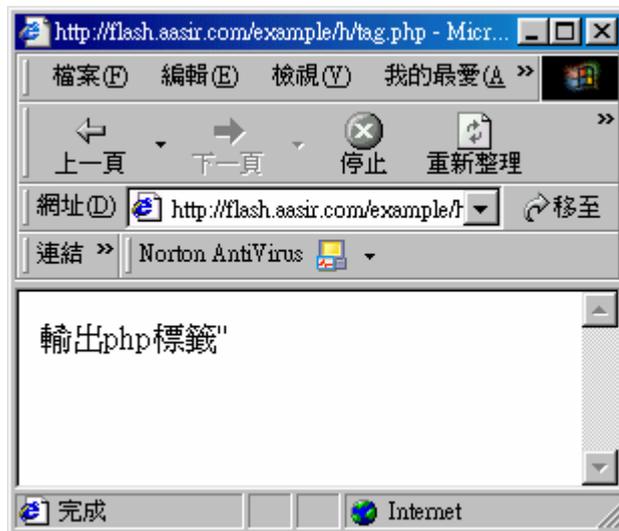


1-1-1php 標籤

<?php 為 PHP 動態網頁標籤起始的標籤，?>為動態 PHP 網頁結束的標籤。和 HTML 的標籤一樣，HTML 網頁起始是<html>標籤，而結束是</html>標籤。在 <?php 和?>標籤中間我們可以撰寫 PHP 的程式，而裏面也可以包含 HTML 的網頁標籤。

範例：tag.php

```
1 <?php
2 print("輸出php標籤 '<?php ?>'");
3 ?>
```



1- 1-2 輸出

我們可以使用 print 來輸出字串或變數到螢幕。我們也可以使用 echo 來輸出字串與變數。

範例：print.php

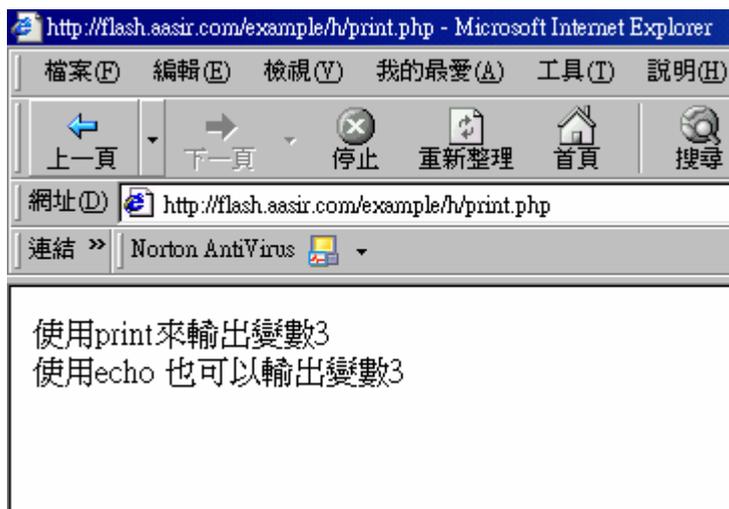
第二行將 3 的數值給變數\$variable。

第三行 print 列印出括號內的字串” 使用 print 來輸出變數\$variable
”，
在網頁中是表示換行的意義。

第四行使用 echo 顯示”使用 echo 也可以輸出變數\$variable”。

```
1 <?php
2 $variable=3;
3 print("使用print來輸出變數$variable<br>");
4 echo "使用echo 也可以輸出變數$variable";
5 ?>
```

因為使用
網頁換行符號，所以顯示兩行。



1-1-3 一行的結束

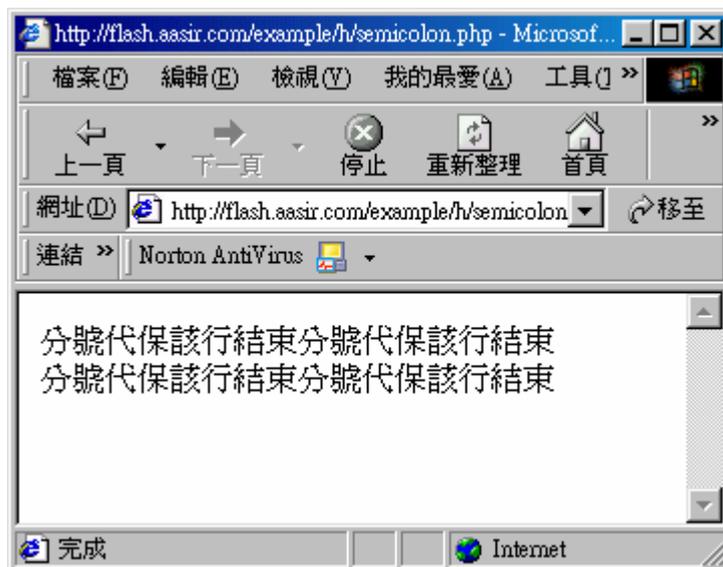
我們使用 ; 分號來代表在 PHP 中該行已經結束。

範例：semicolon.php

第二行第二個 print 代表著已經是另外一行的程式，因為前面有 ; 分號。

第二行的兩個 print 所輸出的結果和第三行與第四行輸出的結果是相同的，雖然第二行的第二個 print 是寫在第二行的 ; 分號後面，其和換行寫是相同的意義。

```
1 <?php
2 print("分號代保該行結束");print("分號代保該行結束<br>");
3 print("分號代保該行結束");
4 print("分號代保該行結束<br>");
5 ?>
```



1-1-4html 標籤

PHP 可以將 HTML 的標籤包含在裏面來使用，最常見的為換行
標籤，這代表在網頁
標籤後為換另外一行。

範例：html.php

第二行將 HTML 網頁的超連結標籤包含到 PHP 的內容裏面

第三行將 HTML 網頁的粗體標籤放到 PHP 的內容裏面。

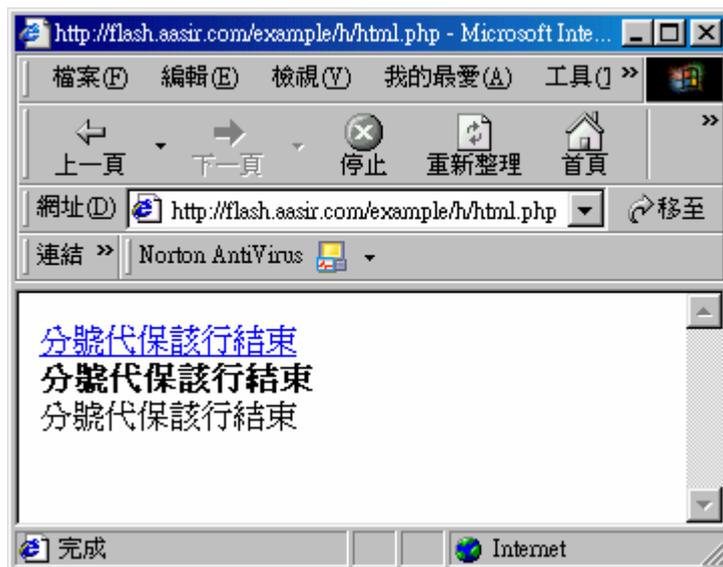
為 HTML 網頁的換行標籤。

```
1 <?php
2 print("<a href='aasir.com'>分號代保該行結束</a><br>");
3 print("<b>分號代保該行結束</b><br>");
4 print("分號代保該行結束<br>");
5 ?>
```

將 HTML 網頁的超連結標籤包含到 PHP 的內容裏面，所以第一行有超連結的作用。

將 HTML 網頁的粗體標籤放到 PHP 的內容裏面，所以第二行為粗體字。

為 HTML 網頁的換行標籤，所以有三行。

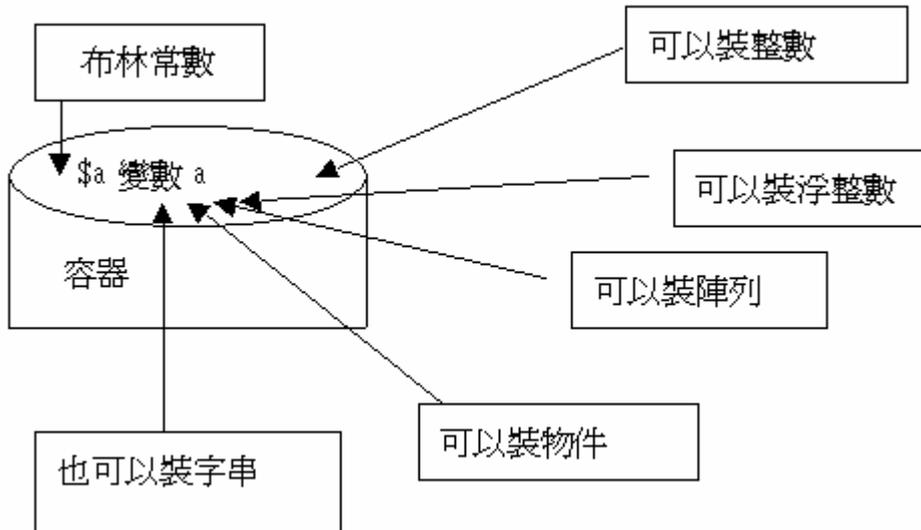


1-2 資料型態

php 支援的資料型態有，陣列(array)、浮整數(floating point)、整數(integer)、物件(object)、布林常數(Boolean)、空值(null)和字串(string)。

\$這個錢符號為變數的符號，例如\$a 就是變數 a 的意義。

這就像是容器可以裝很多東西一樣。



1-2-1 整數(integer)

一般我們都是以十進位為基底，這個變數 a 可以是正整數也可以是負整數。整數就是 $Z=\{\dots, -2, -1, 0, 1, 2, \dots\}$ 的集合。整數可以用十六進位、十進位或八進位來表示，前面可以加正負符號。整數的範圍是從 2147483647 到 -2147483647。

範例 integer.php

```
1 <?php
2 $a = 1234; # 十進位
3 echo $a;
4 echo "<br>";
5 $a = -123; # 負數
6 echo $a;
7 echo "<br>";
8 $a = 0123; # 八進位 (等於十進位的83)
9 echo $a;
10 echo "<br>";
11 $a = 0x12; # 十六進位 (等於十進位的18)
12 echo $a;
13 ?>
```

在瀏覽器所顯現的都會自動轉換成十進位。



範例 : largest_integer.php

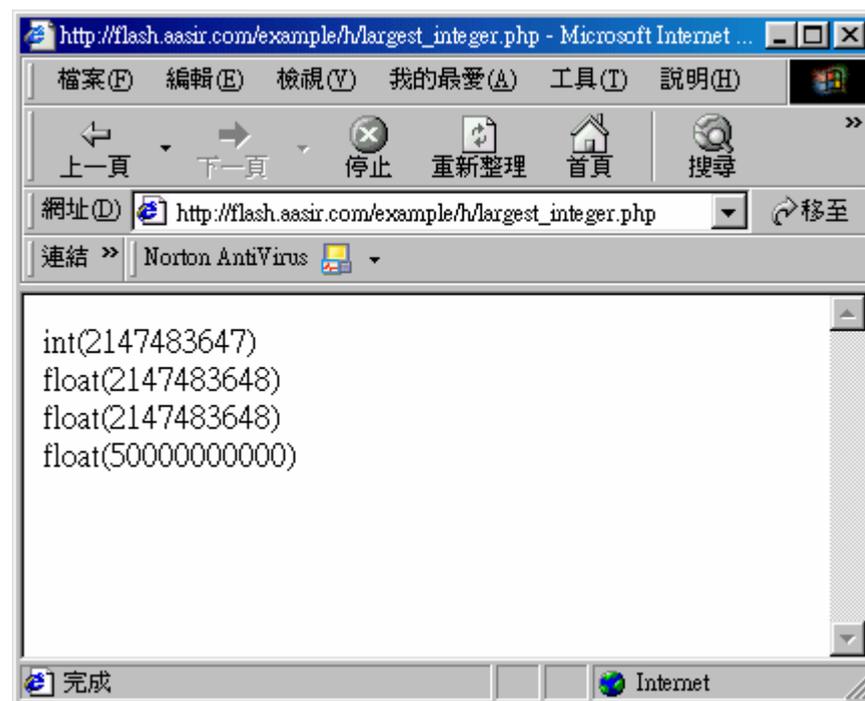
整數 integer 的範圍從 $(2^31-1)2147483647$ 到 $-(2^31-1)-2147483647$, 如果超過就會變成浮點數。

第二行的變數還在整數的範圍內, 所以執行第三行時就可以倒出它的整數值 int。
第五行的變數因為已經超過整數的範圍, 所以執行時會以浮點數 float 的型態來表示。

第八使用 var_dump(0x80000000)來倒出十六進位的符點數。

第十二行也是浮點數, 因為它的數值已經超過整數的範圍了。

```
1 <?php
2 $large_number = 2147483647;
3 var_dump($large_number);
4 echo '<br>';
5 $large_number = 2147483648;
6 var_dump($large_number);
7 echo '<br>';
8 var_dump( 0x80000000 );
9 echo '<br>';
10 $million = 1000000;
11 $large_number = 50000 * $million;
12 var_dump($large_number);
13 echo '<br>';
14 ?>
```



1-2-2 浮點數(floating point number)

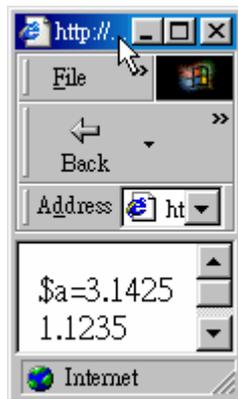
浮點數就是有小數的數值，例如 3.1425 這就是浮點數。浮點數的範圍比整數的範圍還大，在 IEEE 的 64 位元格式中，浮點數的值可以到 $1.8e308$ (10 的 308 次方)，浮點數精確度可以到小數點第十四位。

範例：float.php

第二行的“\”符號為跳脫字元。“\n”為換行符號。

```
1 <?php
2 echo "\$a=3.1425\n";
3 echo $a=1.1235;
4 ?>
```

顯示出小數



範例：float_range.php

第二行 $1e1$ 為 1 乘以 10 的 1 次方，也就是 10。

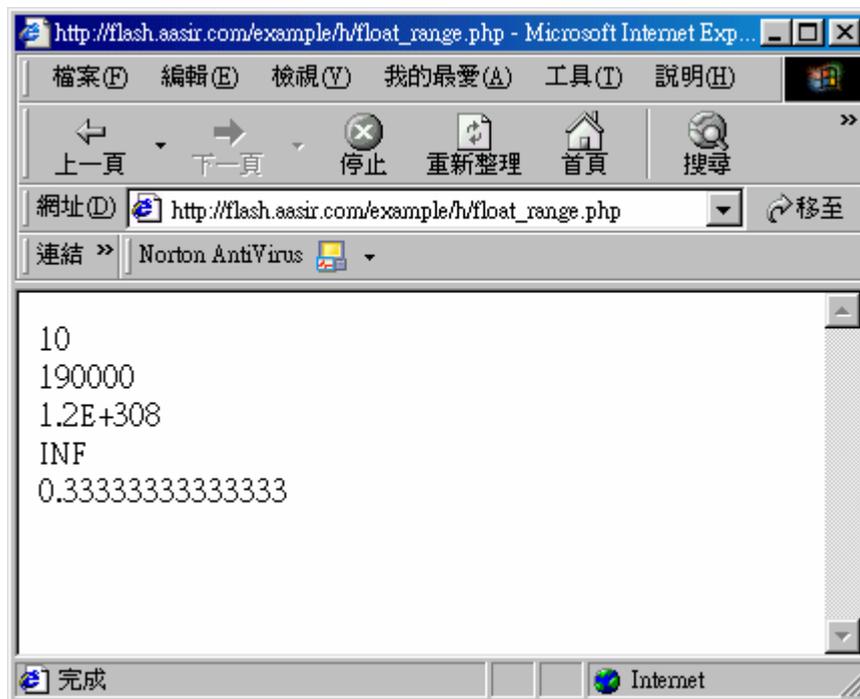
第四行 $1.9e5$ 為 1.9 乘以 10 的 5 次方，也就是 190000。

第六行 $1.2e308$ 為 1.2 乘以 10 的 308 次方。

第八行 $1.9e308$ 已經超過浮點數的範圍，所以顯示 inf 無限大。

第十行將 1 除以 3，所以得到浮點數 0.33333333333333 到小數點十四位數。

```
1 <?php
2 $a=1e1;
3 echo $a.'  
';
4 $a = 1.9e5;
5 echo $a.'  
';
6 $a = 1.2e308;
7 echo $a.'  
';
8 $a = 1.9e308;
9 echo $a.'  
';
10 $a = 1/3;
11 echo $a.'  
';
12 ?>
```



1-2-3 字串(strings)

字串就是用"..."(雙引號)這個符號包起來的內容。字串就是由一連串的字元所組成。PHP 沒有充份的支援 Unicode。一個字元就像是一個位元組，而總共有 256 個不同的字元可以讓我們選擇。字串是沒有長度限制的。

我們可以使用單引號'、雙引號"、heredoc 來指定字串。

範例 strings.php

我是大好人就是字串，它的前後都有雙引號"。

先將字串給\$a 這個變數，再用 echo 來顯示\$a 這個變數。

```
1 <?php
2 $a="我是大好人";
3 echo $a;
4 ?>
```



範例：single_quote.php

\符號為跳脫字元。

第一行我們將字串包含在單引號'內。

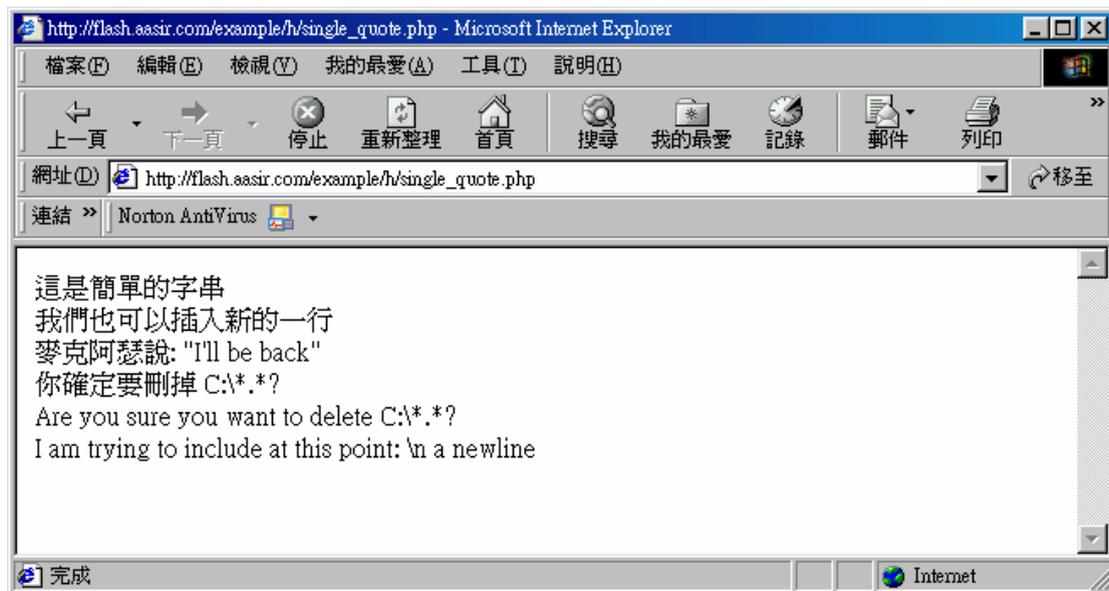
第二行和第三行，我們將資料也是用單引號包住。

第五行我們使用跳脫字元讓單引號\沒有作用，只顯示單引號。

第六行我們使用跳脫字元讓\跳脫字元沒有作用，只顯示\反斜線。

```
1 <?php
2 echo '這是簡單的字串<br>';
3 echo '我們
4 也可以插入新的一行<br>';
5 echo '麥克阿瑟說："I\'ll be back"<br>';
6 echo '你確定要刪掉 C:\\*..*?<br>';
7 echo 'Are you sure you want to delete C:\\*..*?<br>';
8 echo 'I am trying to include at this point: \n a newline';
9 ?>
```

這是顯示的情況。



範例：heredoc.php

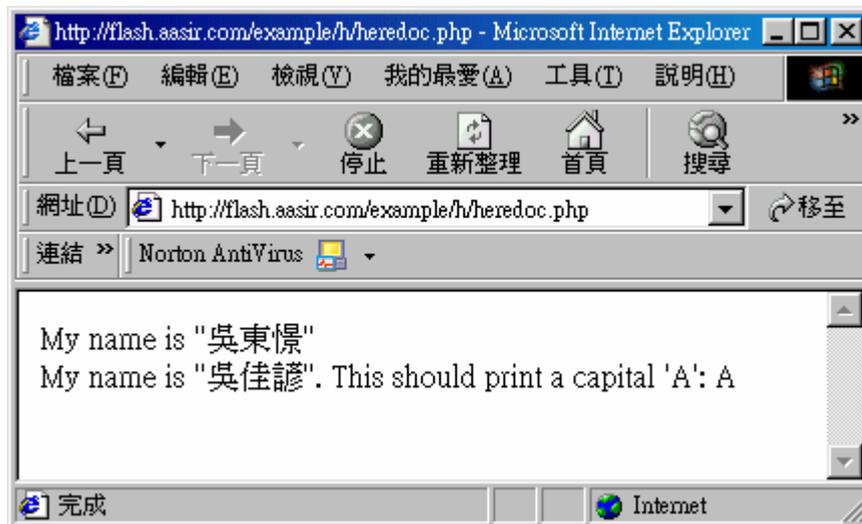
<<<為 heredoc 符號。在 heredoc<<<符號後面加一個識別子符號，在字串結尾再加上該識別子符號。

在第三行<<<為 heredoc 符號，EOD 為我們設定的識別子符號，第五行 EOD 為結尾的識別子符號，這樣就可以將字串包在中間了。

第七行<<<為 heredoc 符號，EOT 為我們設定的識別子符號，第十行 EOT 為結尾的識別子符號，這樣就可以將字串包在中間了。

```
1 <?php
2 $name='吳東憬';
3 echo <<<EOD
4 My name is "$name"<br>
5 EOD;
6 $name='吳佳諺';
7 echo <<<EOT
8 My name is "$name".
9 This should print a capital 'A': \x41
10 EOT;
11 ?>
```

這是顯示的結果。

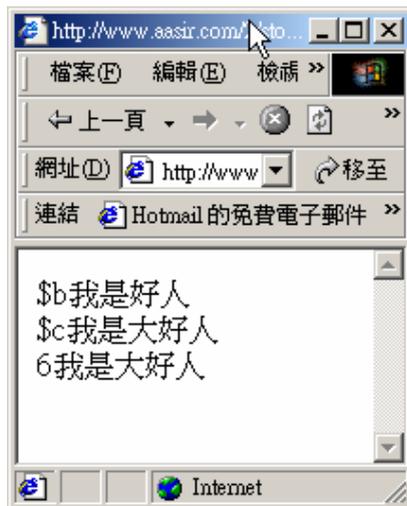


1-2-4 跳脫字元(escape)\

範例：sto.php

單引號內所包含的字串都會顯示。'\$'錢的符號前面需加跳脫字元\"才能顯示出來。我們在第9行使用字串符號\".\"就能將變數\$c 的值和字串相連了。

```
1 <?php
2 $a=5;
3 $b=3;
4 $c=6;
5 echo '$b我是好人';
6 echo "<br>";
7 echo "\$c我是大好人";
8 echo "<br>";
9 echo $c."我是大好人";
10 ?>
```



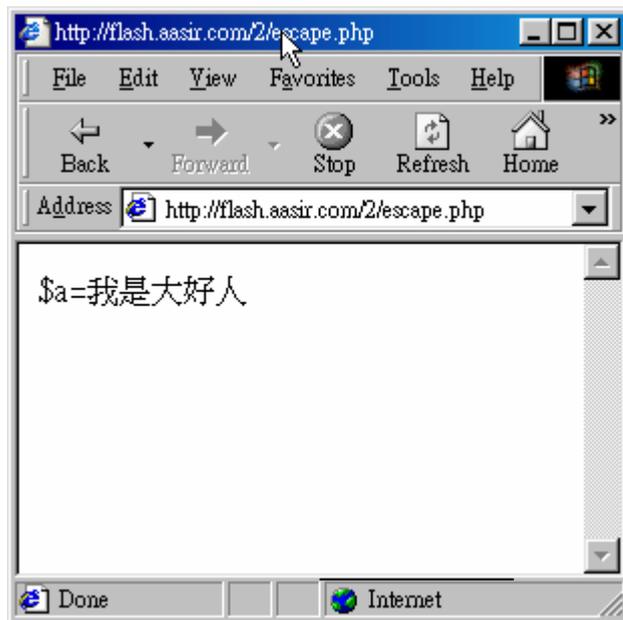
跳脫字元	說明
\"	將雙引號\"當作字元
\n	換行
\\$	將\$當作字元
\\	將\"當作字元
\x00 到\xFF	以十六進位表示某一位元
\000 到\377	以八進位表示某一位元

範例：escape.php

我們在第二行使用跳脫字元\`$a` 來顯示\$。

```
1 <?php
2 $a="\ $a=我是大好人<br>";
3 echo $a;
4 ?>
```

\跳脫字元前面接\$錢的符號，這樣就可以顯示這個\$的符號。



範例：escape1.php

在字串符號上，加上
標籤，這樣就可以換行了。



1-2-5 布林常數(Boolean)

布林常數的值不是真 true 就是假 false 這兩種 當值為 1 時為 true , 為 0 或空值(null) 時為 false。

範例 : boolean.php

第二行將 True 的值給變數\$a。

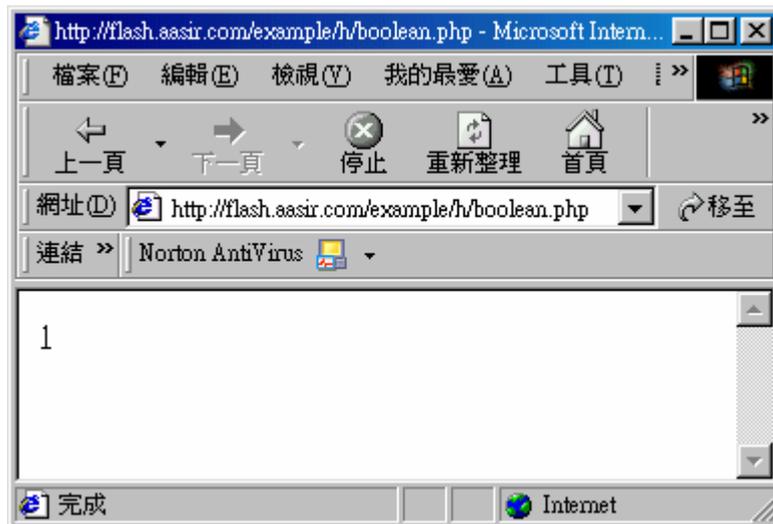
第三行顯示變數\$a 的值為 1。

第五行將 false 的值給變數\$b。

第六行顯示變數\$b 的值為空值(null value) , 就是沒有顯示資料的意義。

```
1 <?php
2 $a = True;
3 echo $a;
4 echo '<br>';
5 $b = false;
6 echo $b;
7 ?>
```

執行的結果只有顯示 1 , 其它的則沒有顯示 , 因為 false 為空值。



範例：boolean_test.php

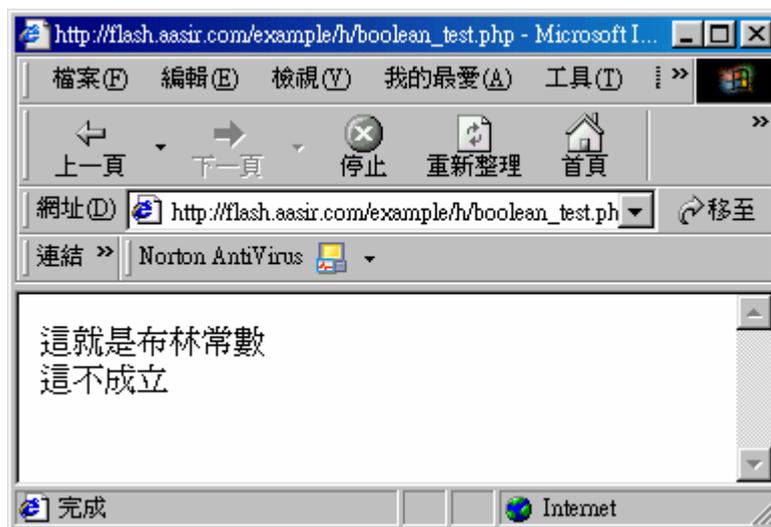
布林常數(Boolean)經常用在判別式。當真(true)時，就會執行某一段結構，當為假(false)時，就會執行假所選定的結構。

第二行將布林常數的 true 給變數\$a，因此在第三行判別時為真(true)，所以會執行第四行，將"這就是布林常數"顯示出來。

第九行將布林常數的 false 給變數\$a，因此在第十行判別時為假(false)，所以執行第十三行，將"這不成立"顯示。

```
1 <?php
2 $a=true;
3 if($a)
4     echo "這就是布林常數";
5 else
6     echo "這不成立";
7
8 echo '<br>';
9 $a=false;
10 if($a)
11     echo "這就是布林常數";
12 else
13     echo "這不成立";
14 ?>
```

這是執行的情況。



1-2-6NULL

NULL 就是空值(空空的)的意義。NULL 的型態為 NULL。

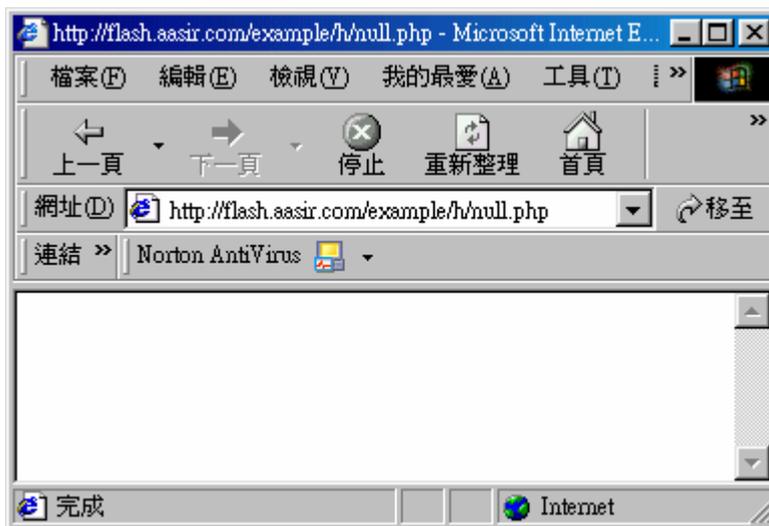
範例：null.php

第二行將 null 空值給變數\$a。

第三行顯示空值，就是沒有顯示資料。

```
1 <?php
2 $a=null;
3 echo $a;
4 ?>
```

執行的結果沒有顯示資料。



1-2-7 陣列(arrays)

陣列就是一個連續的空間放著各種資料。

例如下面是一段記憶體的位址。

0	你好
1	我好
2	大家好
3	123
4	就有
5	0.567
6	thanks

它表現的方法就像下面的。

第二行將'你好'放到陣列 a 在 0 的位址。

第三行將'我好'放到陣列 a 在 1 的位址。

第四行將'大家好'放到陣列 a 在 2 的位址。

第五行將整數 123 放到陣列 a 在 3 的位址。

第六行將字串'就是'放到陣列 a 在 4 的位置。

第七行將浮整數放到陣列 a 在 5 的位址。

範例練習 array.php

```
1 <?php
2 $a[0]="你好";
3 $a[1]="我好";
4 $a[2]="大家好";
5 $a[3]=123;
6 $a[4]="就是";
7 $a[5]=0.567;
8 $a[6]="thanks";
9 ?>
```

範例練習 array1.php

範例 array.php 的 a[]陣列可以表示成第二行的表示法。

```
1 <?php
2 $a=array("你好","我好","大家好",123,"就是",0.567,"thanks");
3 ?>
```

<<1>>二維陣列

下面就是一個 2 維陣列\$a，第一列我們存放著 4 個變數，第二列我們也存放著 4 個變數。

第 0 列第 0 行	第 1 欄	第 2 欄	第 3 欄	第 4 欄
第 1 列	1	2	3	5
第 2 列	6	7	8	9

\$a[1][1]=1;將 1 這個數放到陣列\$a 第一列第一行

\$a[1][2]=2;

\$a[1][3]=3;將 3 這個數放到陣列\$a 第一列第三行

\$a[1][4]=5;將 5 這個數放到陣列\$a 第一列第四行

\$a[2][1]=6;

\$a[2][2]=7;

\$a[2][3]=8;將 8 這個數放到陣列\$a 第二列第三行

\$a[2][4]=9;將 9 這個數放到陣列\$a 第二列第四行

範例 array5.php

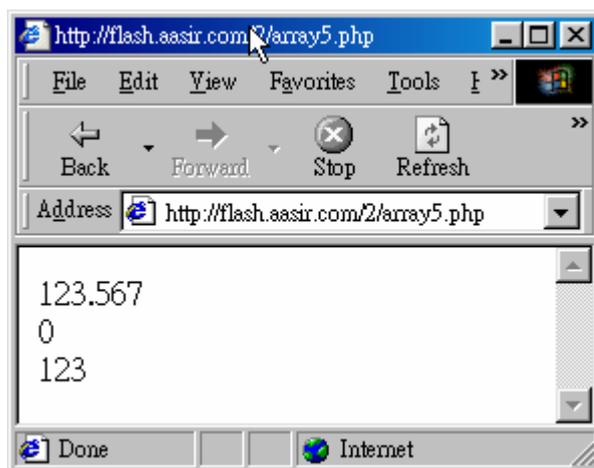
字串作數值運算時會變成整數 0。

第九行中陣列\$a 第 3 個數值 123 加陣列\$a 第 5 個數值 0.567 共為 123.567。

第十一行字串相加為 0。

第十三字串加數值 123 共為 123。

```
1 <?php
2 $a[0]="你好";
3 $a[1]="我好";
4 $a[2]="大家好";
5 $a[3]=123;
6 $a[4]="就是";
7 $a[5]=0.567;
8 $a[6]="thanks";
9 echo $a[3]+$a[5];
10 echo "<br>";
11 echo $a[0]+$a[2];
12 echo "<br>";
13 echo $a[0]+$a[3];
14 ?>
```



範例 arrays.php

我們可以將陣列裏面的值作計算。

```
1 <?php
2 $a[1][1]=1;
3 $a[1][2]=2;
4 $a[1][3]=3;
5 $a[1][4]=5;
6 $a[2][1]=6;
7 $a[2][2]=7;
8 $a[2][3]=8;
9 $a[2][4]=9;
10 echo $a[1][3]+$a[2][3];
11 ?>
```

將陣列\$a 第一列第三行的值 3 加上第二列第三行的值 8 就會等於 11。



<<3>>多維陣列

我們可以用一維陣列，二維陣列，當然也可以用多維陣列。

範例 arraym.php

第七行字串相加為 0。

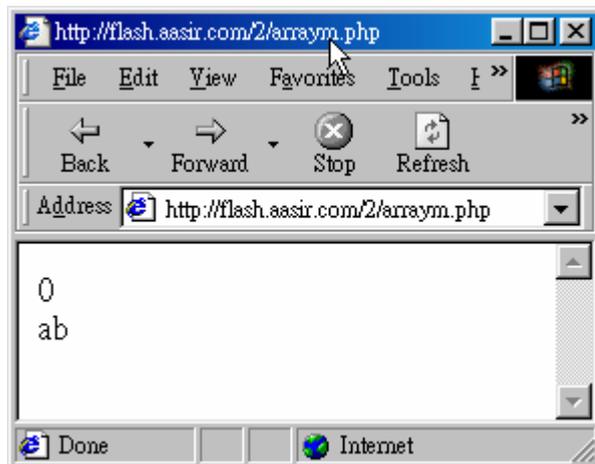
第八行為換行標籤

第九行為合併符號逗點'.'

```
1 <?php
2 $a[0][0][0]="a";
3 $a[0][0][1][0]="b";
4 $a[0][0][1][1]="c";
5 $a[0][0][1][1][1]="d";
6 $a[0][1][2][3]="d";
7 echo $a[0][0][0]+$a[0][0][1][0];
8 echo "<br>";
9 echo $a[0][0][0].$a[0][0][1][0];
10 ?>
```

字串相加為 0。

使用字串合併符號就可顯示 ab 連在一起了。



1-3 物件

物件也是資料形態的一種。

類別是抽象化(abstraction)的，而物件就是實體化。對電腦而言，類別只是一些程式碼，而真正在記憶體上配置空間的是物件，也就是說物件才是我們互動的資料。

物件導向使我們寫程式更為方便，雖然 php 不像 C++ 或 JAVA 一樣這麼完整的設計，但對於 script 來說已經足夠了。當我們要使用之前開發過的程式(基礎類別 base class)，然後要改成目前計畫的程式，只要將它的類別繼承再作修改或增加，成為衍生類別(derived class)了，php 只支援單一繼承，也就是說一個類別只能繼承另外一個類別，就比較像 Java。如果要將類別實體化，則要用 new 這個動作，來配置記憶體給這個物件，也就實體化了。物件導向還有一項優點，在類別裏面，有著許多的資料和方法，而只有在類別裏才看得到，這個就稱為封裝(encapsule)。

範例 nu.php

在 nu.php 中我們將線的長度及三角形面積及正方形面積的方法都隱藏在 a 這個類別中，這個 a 類別是抽象的當我們要使用它時就須要配置記憶體給它，使它成為物件 area，再用->這個符號去取(指向)它的屬性或方法。

我們在第二行宣告類別 a。

第三行為類別 a(class a)的變數\$line。

第四行到第六行為三角函數 function triangle 的宣告。

第七行到第九行為正方形函數 function squre 的宣告。

第十一行使用 new 來實體化類別 a。

```
1 <?php
2 class a { //a為處理面積的類別
3     var $line=100; //線的長度100
4     function triangle ($x, $y) { //三角形面積方法
5         return ($x*$y)/2;
6     }
7     function squre($z) { //正方形面積方法
8         return ($z*$z);
9     }
10 }
11 $area = new a; //為類別a配置計憶體並且讓$area指向它
12 echo $area->line; //顯示面積物件的線
13 echo "<br>";
14 echo $area->triangle(1,2); //顯示三角形的面積
15 echo "<br>";
16 echo $area->squre(3); //顯示正方形的面積
17 ?>
```

線的長度為 100。三角形的面積為 1。

底(1)*高(2)/2=面積(1)。正方形面積 9。



1-4 形別轉換 Type Casting

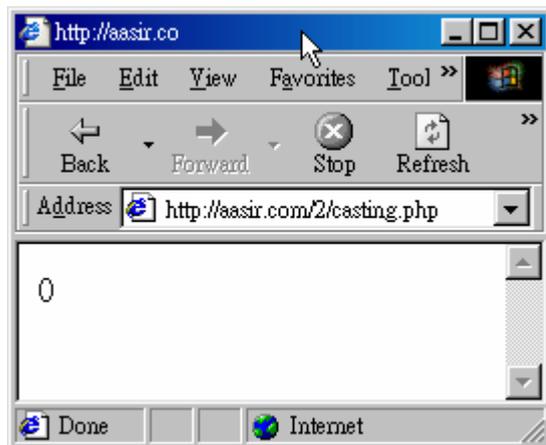
php 並不支援清楚的型態定義。在 c 語言裡，整數是用 int 的形態、字元是用 char 的形態、浮點數是用 float 的形態，在 php 裏頭變數都是 var 來表示，當要用到時的狀況，php 會自動決定變數的形態。例如我們將兩個字串相加(+)時，所得的就是 0 的數值，因為字串作數值運算時，php 會自動把字串轉為數值 0，相加的值就變成 0 了。

範例:casting.php

在第四行字串相加時 php 會把字串形態轉為數值 0 再相加。

```
1 <?php
2 $a="我";
3 $b="要";
4 $c=$a+$b;
5 echo $c;
6 ?>
```

\$a+\$b 的數值顯示為 0。



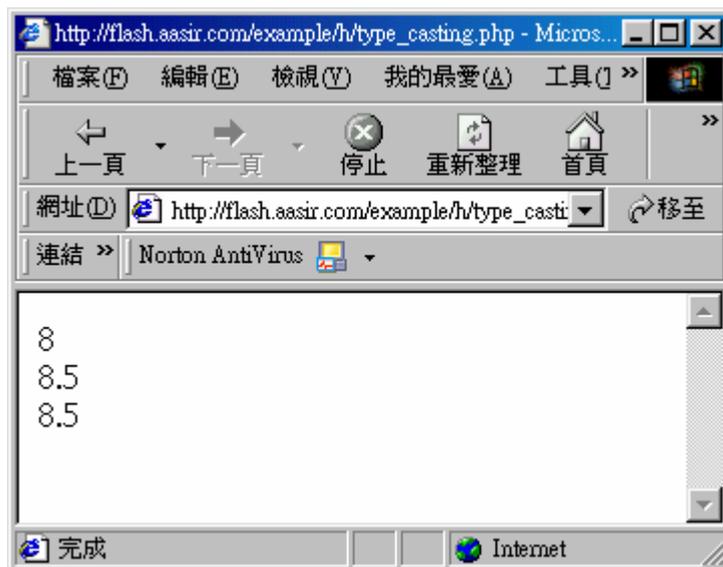
範例：type_casting.php

第三行將浮點數轉換成整數 int，所以顯示 8。

第六行顯示浮點數，所以顯示 8.5。

第九行將浮點數轉換成字串型態。

```
1 <?php
2 $data=8.5;
3 $data=(int) $data;
4 echo $data.'<br>';
5 $data=8.5;
6 $data=(double) $data;
7 echo $data.'<br>';
8 $data=8.5;
9 $data=(string) $data;
10 echo $data.'<br>';
11 ?>
```



1- 5 變數

在 php 中變數的表示就是前面加一個\$錢的符號，後面加上名子。名子的寫法可以第一個字是大小寫英文字也可以是底線，但第一個字不可以為數字。

第一個字元從 a-z、 A-Z 和 ASCII 字元從 127 到 255 (0x7f-0xff).

範例：de.php

變數的名稱為 var，可以。

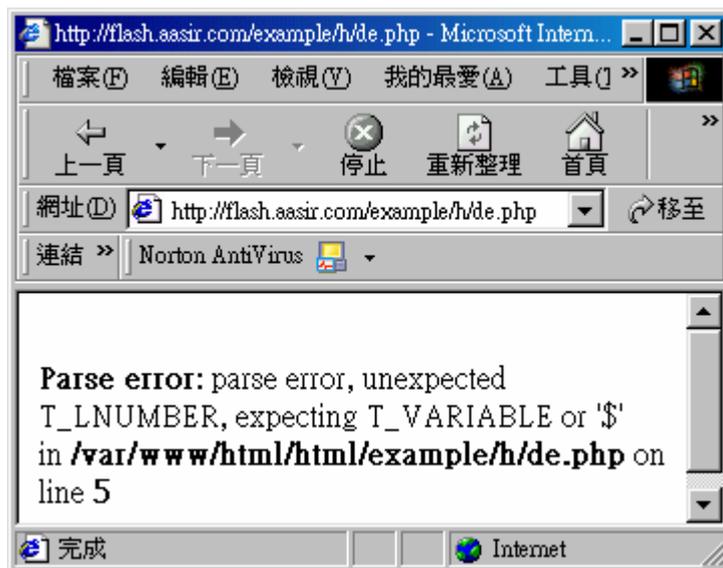
變數名稱 4site 第一個字為 4，不可以。

第一個字為底線，可以。

第五行及第六行宣告及使用不合法的變數\$4site，因為使用數字為開頭的變數。

```
1 <?php
2 $var = "吳佳諺";
3 $Var = "吳東憬";
4 echo "$var, $Var". '<br>';
5 $4site = '可不可以當變數';
6 echo $4site. '<br>';
7 $_4site = '可以當變數';
8 echo $_4site;
9 ?>
```

這時只要執行就會發生語法錯誤。



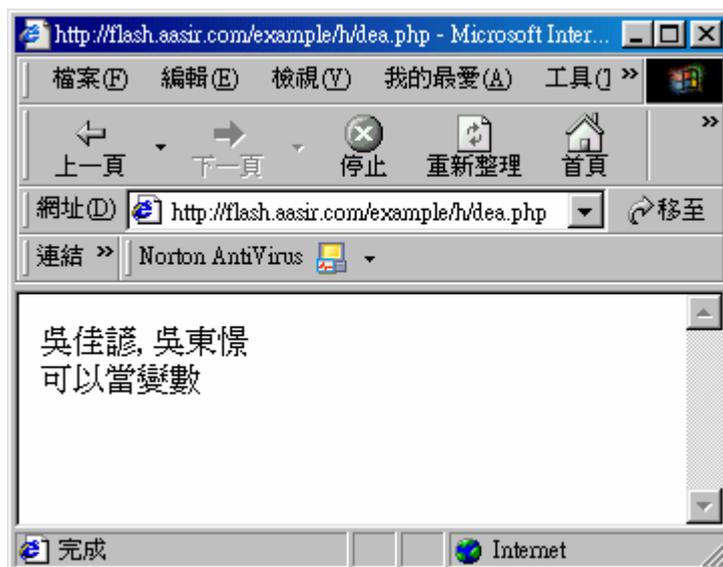
範例：dea.php

將第五行和第六行去除後，就可以顯示資料了。

第七行使用_底線為開頭的變數。

```
1 <?php
2 $var = "吳佳諺";
3 $Var = "吳東憬";
4 echo "$var, $Var". '<br>';
5 // $4site = '可不可以當變數';
6 // echo $4site. '<br>';
7 $_4site = '可以當變數';
8 echo $_4site;
9 ?>
```

這是顯示的情況。



1-5-1&參考

&參考符號要是在變數的面前指的就是變數本身的位址。

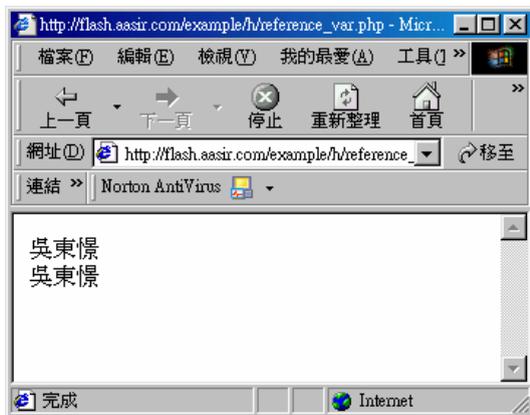
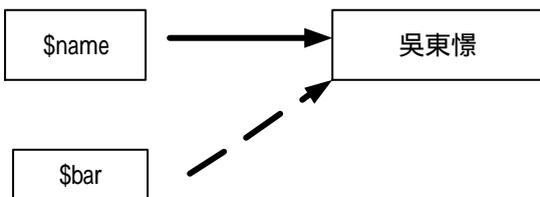
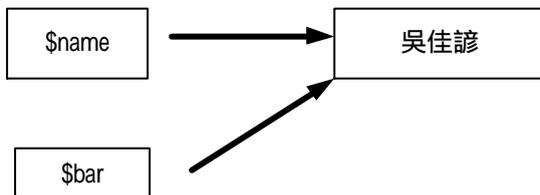
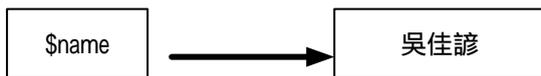
第二行將字串”吳佳諺”給\$name 變數。

第三行將\$bar 變數指向字串”吳佳諺”，也就是參考&變數\$name。

第四行將\$bar 變數的內容改成吳東憬。

第五行顯示\$bar 的內容”吳東憬”，因為\$bar 參考&到\$name 變數，當\$bar 的內容改變時，\$name 所參考的內容因為是和\$bar 相同，所以\$name 的內容也就是\$bar 的內容。。

```
1 <?php
2 $name = '吳佳諺';
3 $bar = &$name;
4 $bar = "吳東憬";
5 echo $bar.'<br>';
6 echo $name;
7 ?>
```

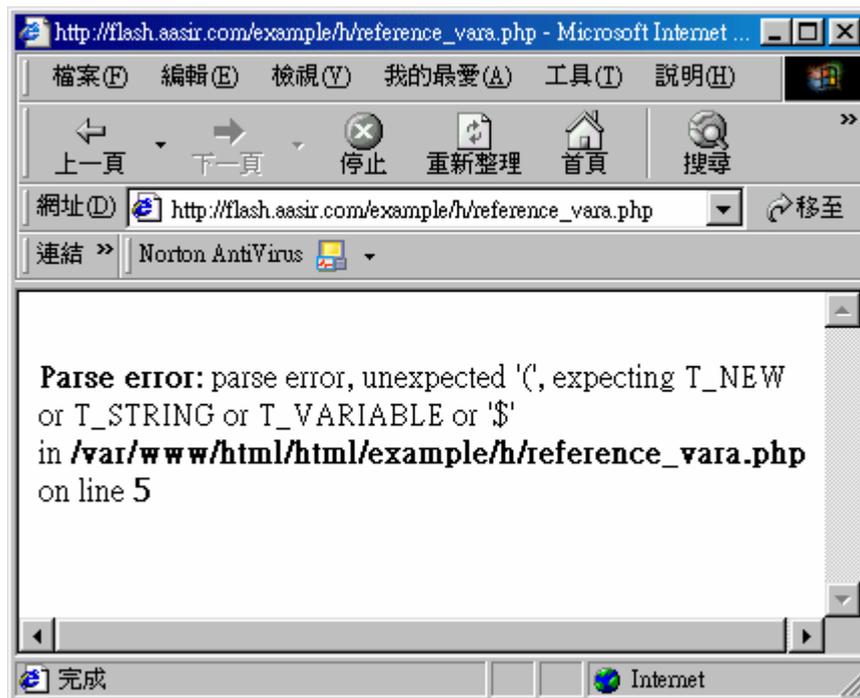


範例：reference_vara.php

第五行和第十一行錯誤，因為參考&是針對變數，而不是一般的數值。

```
1 <?php
2 $foo = 25;
3 $bar = &$foo;
4 echo $bar.'<br>';
5 $bar = &(24 * 7);
6 echo $bar.'<br>';
7 function test()
8 {
9     return 25;
10 }
11 $bar = &test();
12 echo $bar;
13 ?>
```

顯示語法錯誤。



1-6 保留字 predefined variables

php 保留了大量的保留字，有些是和 apache 伺服器有關，有些則是 php 自己的保留字，保留字，就是 php 以經定義過的字，它有自己的功用。

1-6-1 apache 保留字

這些保留字的變數各顯示出各自的資訊，例如\$server_name 保留字顯示出伺服器的 ip，而\$remote_addr 顯示出瀏覽者的 ip，而\$http_host 顯示了，我們網站的名子。

範例 predefined.php

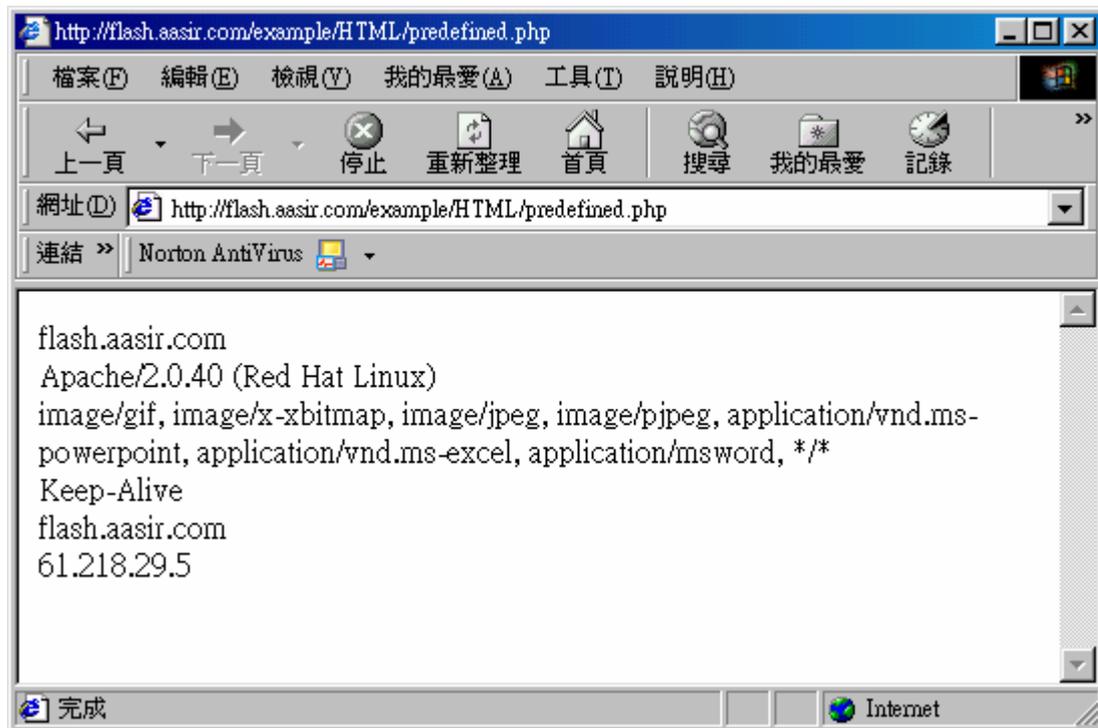
```
1 <?php
2 echo $SERVER_NAME;
3 echo "<br>";
4 echo $SERVER_SOFTWARE;
5 echo "<br>";
6 echo $HTTP_ACCEPT;
7 echo "<br>";
8 echo $HTTP_CONNECTION;
9 echo "<br>";
10 echo $HTTP_HOST;
11 echo "<br>";
12 echo $REMOTE_ADDR;
13 ?>
```

伺服器的軟體：Apache

保持連線：keep-Alive

主機的名子：flash.aasir.com

使用者的 ip：61.218.29.5



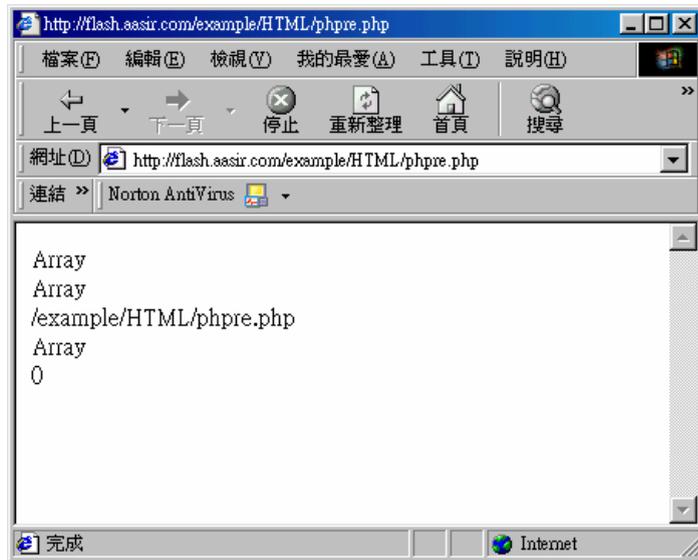
1-6-2php 保留字

php 的保留字有 \$php_self 顯示目前相對於根目錄的檔案路徑。

範例練習 phpre.php

```
1 <?php
2 echo $HTTP_SERVER_VARS;
3 echo "<br>";
4 echo $HTTP_COOKIE_VARS;
5 echo "<br>";
6 echo $PHP_SELF;
7 echo "<br>";
8 echo $argv;
9 echo "<br>";
10 echo $argc;
11 ?>
```

這是 php 保留字所顯示的資訊。



1-7 變數生存空間 global variables

變數的作用範圍，依據變數所在的空間。

全域變數作用的範圍為整個檔案，其變數所在的地方為整個檔案；區域變數作用的範圍在函數裏面，其變數所在空間為函數裏面

1-7-1 全域變數

作用範圍為整個檔案。

範例第三章 scope.php、b.fun

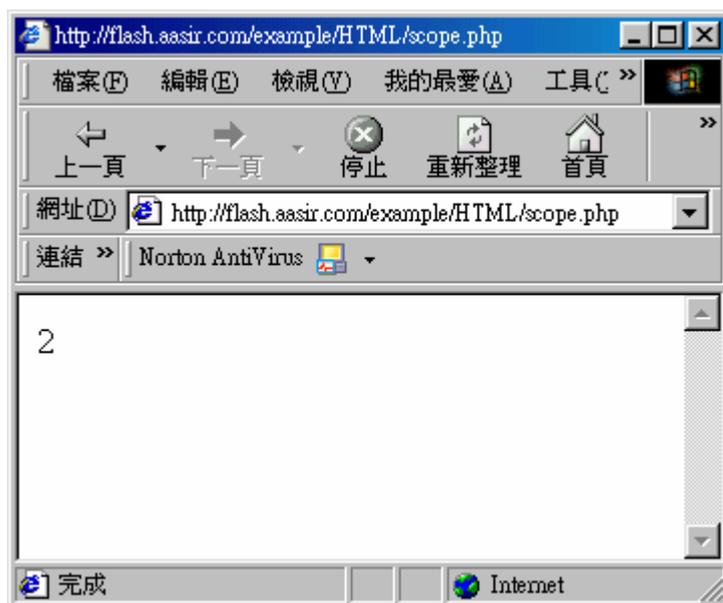
這是 scope.php 的檔案，變數\$a 在這邊是全域變數，也就是說它的作用範圍為全部區域。

```
1 <?php
2 $a=2;
3 include "b.fun";
4 ?>
```

這是 b.fun 的檔案，因為 scope.php 把 b.fun 檔包含進來，所以變數\$a 也在其裏面作用。

```
1 <?php
2 echo $a;
3 ?>
```

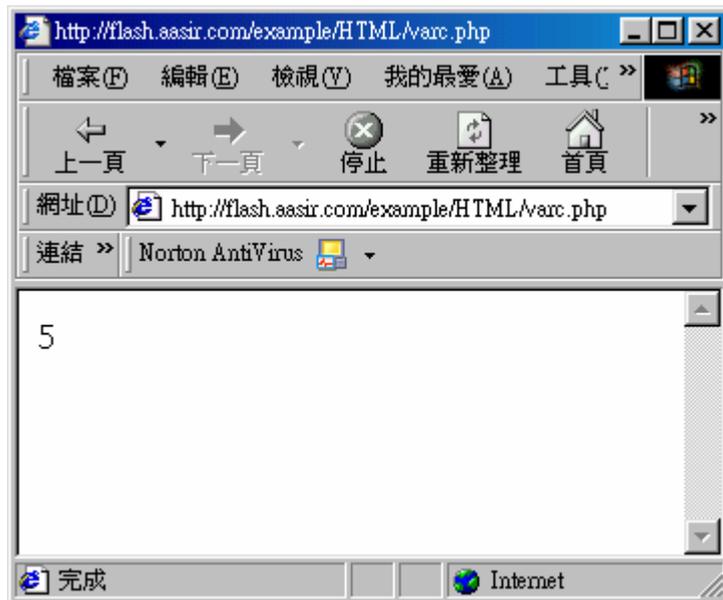
在 b.fun 檔中顯示變數\$a 為 2



範例 varc.php

當我們想在函數裡使用全域變數時可使用\$GLOBALS 陣列，因為我們全域變數都放在\$GLOBALS 中，在函數中我們使用變數的名子當作\$globals 陣列的索引 \$global[name]。

```
1 <?php
2 $a=5;
3 function gl( )
4 {
5   $a=$GLOBALS[a];
6   echo $a;
7 }
8 gl( );
9 ?>
```



範例 varb.php

在函數裡面如果要使用全域變數，除了使用陣列\$GLOBALS 外，只要在函數裏面使用 global的宣告，就可以知道這是全域變數了。

我們在第三行宣告全域變數\$a 和全域變數\$b。

```
1 <?php
2 $a = 1;
3 $b = 2;
4 Function Sum () {
5     global $a, $b;
6     $b = $a + $b;
7 }
8 Sum ();
9 echo $b;
10 ?>
```



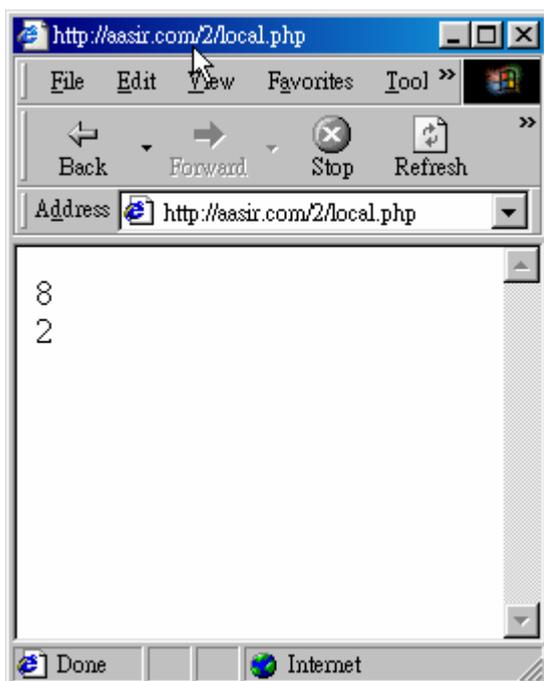
1-7-2 區域變數 local variables

區域變數生存空間與作用範圍局限在函數裏面。

在函數 sum 裏面的變數\$a 和變數\$b 都只有作用在函數裏面(在函數裏的\$a 與\$b 皆為區域變數)，所以顯示第一個變數\$b 時，其值為 8；當顯示第 2 個變數時其值是全域變數\$b 值為 2，所以顯示 2。

```
1 <?php
2 $a = 1;
3 $b = 2;
4 Function Sum () {
5     $a=3;
6     $b=5;
7     $b = $a + $b;
8     echo $b;
9 }
10 Sum ();
11 echo "<br>";
12 echo $b;
13 ?>
```

區域變數值的合為 8，它的作用空間只有在第四行到第九行的 sum()函數內。
全域變數\$b 值 2，它的作用空間是在整個的網頁。



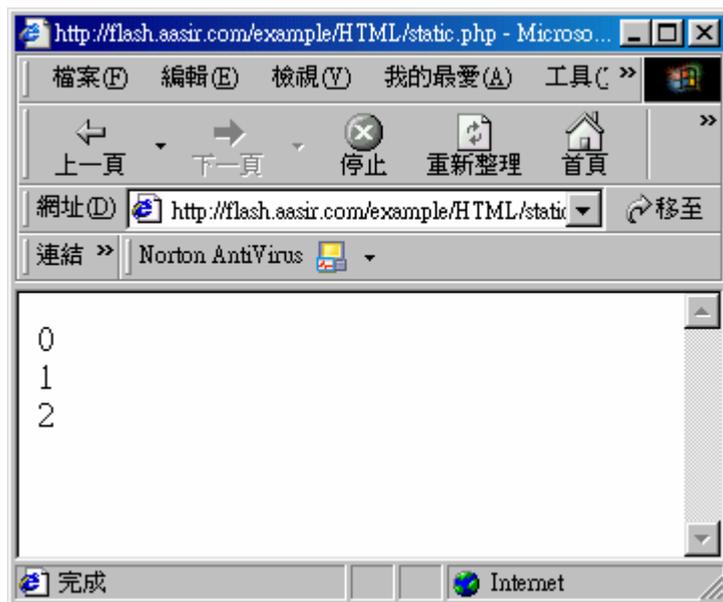
1-7-3 靜態變數(static variable)

靜態變數是只有存在於(函數的區域範圍)local function scope，但他有一個特質，那就是當離開執行函數時，他的值並不會被消滅，所以稱為靜態變數。

範例 static.php

```
1 <?php
2 Function Test () {
3     static $a = 0;
4     echo $a;
5     $a++;
6 }
7 Test ( );
8 echo "<br>";
9 Test ( );
10 echo "<br>";
11 Test ( );
12 ?>
```

因為變數\$a 為靜態變數，當執行完函數時，他的變數值並不會同時消失，而是繼續保存，第二次呼叫函數 Test()時則是將靜態變數\$a=1 顯示出來，再將靜態變數\$a 加 1，第三次呼叫時則將靜態變數\$a=2 顯示出來。

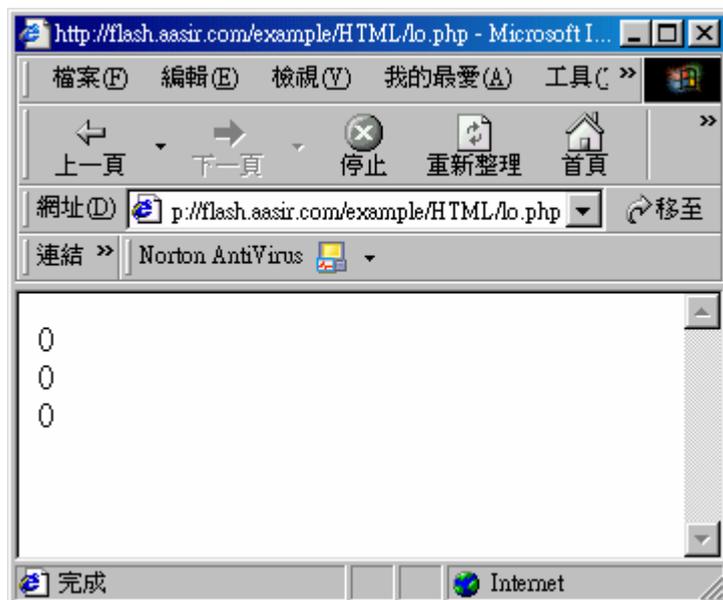


範例 lo.php

在 lo.php 的範例中區域變數\$a，當函數執行過後，區域變數\$a 的值也會消失，所以在第七行和第九行和第十一行執行 test()時會顯示 0。靜態變數是持續存在，而區域變數則隨函數的結束而結束。

```
1 <?php
2 Function Test () {
3     $a = 0;
4     echo $a;
5     $a++;
6 }
7 Test ( );
8 echo "<br>";
9 Test ( );
10 echo "<br>";
11 Test ( );
12 ?>
```

這是執行的結果，顯示 0、0、0。



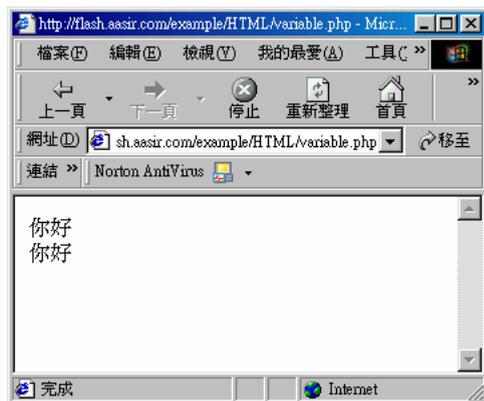
1-7-4 變數的變數(variable variables)

給變數取一個名子是很方便的，例如\$a="hello"我們取一個變數名子為\$a 並將字串"你好"指派給它，如果我們再讓\$hello="你好"，這時字串你好就可以表達成\$\$a。這個變數\$a 就叫字串你好變數的變數(variable variables)。

範例 variable.php

變數\$hello 和變數的變數\$\$a 所顯示的內容一樣都是字串你好。

```
1 <?php
2 $a="hello";
3 $hello="你好";
4 echo $$a;
5 echo "<br>";
6 echo $hello;
7 ?>
```



1-8 外來傳入的變數

當我們登錄網站時要打上帳號和密碼，再按傳送，這時後我們就把資料傳送到網站的伺服器，再經由伺服器端(server)的程式來處理我們傳送的資料。對我們網站來講，這就是從外面傳來的資料與變數，這些資料有些是來自外面的瀏覽器所記錄的 cookies，有一些是我們自己從表單輸入的資料。

1-8-1 從 html 表單傳來的變數

我們經常在網站上使用 email，當然，您的網站要有 sendmail 伺服器才能支援寄信的功能，下面這個範例就是例用網頁的表單來傳送資料給 php 來處理，讓 php 處理後，當作 email 把我們的資料寄出。

範例 sendmail.php

這是網頁傳送資料給自己再處理後由 php 的 mail();函數寄出資料。

Sendmail.php 從 16 行到第 41 行為表格，當我們在表格上輸入文字時，這些文字就會指派給相對應的變數名稱，當我們按下傳送時，這些資料就會傳送到 form action 所指定的地方，在此為自己 sendmail.php。

To：欄位變數名稱為 to。

FROM：欄位變數名稱為 from。

主旨：欄位變數名稱為 subject。

內文：欄位變數名稱為 body

Email 給老師!	
TO:(老師E-MAIL)	<input type="text"/>
FROM:(寄件者E-MAIL)	<input type="text"/>
主旨：	<input type="text"/>
內文： (請包含您的聯絡方式及姓名電話)	<input type="text"/>
<input type="button" value="送出"/>	

第六行使用郵寄函數來寄發 email，變數名稱為 to。

```
1<HTML>
2<BODY bgcolor="#FFFFCC">
3<?php
4if ( $Send ) {
5    if ( $From )
6        mail( $To, $Subject, $Body, "From: $From");
7    else
8        mail( $To, $Subject, $Body);
9    $Msg = "信件已經送出!";
10}
11?>
12<H2 align="center">Email 給家教老師!</H2>
13<H2>
14    <HR>
15</H2>
16<FORM Action=sendmail.php Method=POST>
17<TABLE Border=1>
18    <TR>
19        <TD width="245">TO: (家教老師E-MAILL)</TD>
20        <TD width="368">
21            <INPUT Type=Text Name=To Size=40></TD></TR>
```

第二十六行的變數名稱為 from。

第三十行的變數名稱為 subject。

第三十七行的變數名稱為 body。

```
22    <TR>
23        <TD width="245">FROM: (寄件者E-MAILL)<BR>
24        </TD>
25        <TD width="368">
26            <INPUT Type=Text Name=From Size=40></TD></TR>
27    <TR>
28        <TD width="245">主旨: </TD>
29        <TD width="368">
30            <INPUT Type=Text Name=Subject Size=40></TD></TR>
31    <TR>
32        <TD width="245">
33            <p>內文: </p>
34            <p>(請包含您的聯絡方式及姓名電話)</p>
35        </TD>
36        <TD width="368">
37            <TEXTAREA Name=Body Rows=8 Cols=60></TEXTAREA>
38    </TD></TR>
39</TABLE>
40    <div align="center">
41        <INPUT Type=Submit Value="送出" Name="Send">
42    </div>
```

```
43 </FORM>
44 <?php echo $Msg;?>
45 </BODY>
46 </HTML>
```

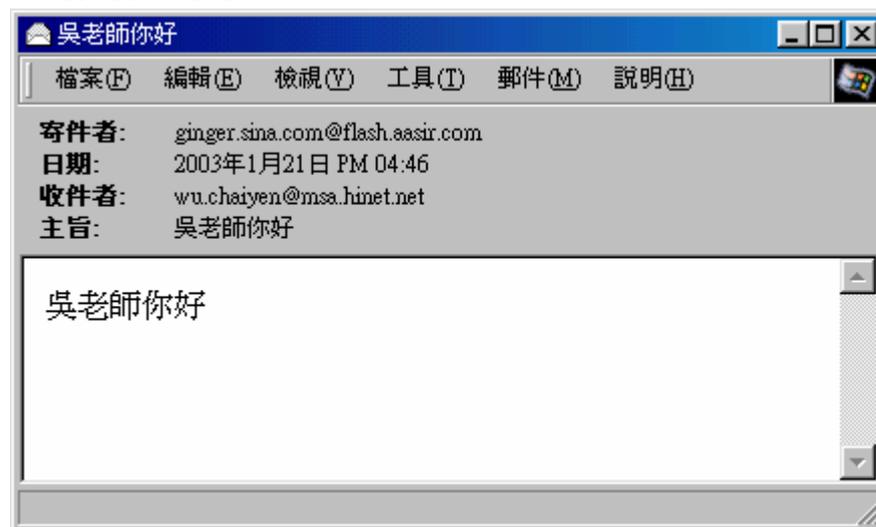
將字串”wu.chaiyen@msa.hinet.net”傳給變數\$to，將字串”ginger.sina.com”傳給變數\$from，將字串”吳老師你好”傳給\$subject，將字串”吳老師你好傳給\$body。

Email 給老師!

TO:(老師E-MAIL)	<input type="text" value="wu.chaiyen@msa.hinet.net"/>
FROM:(寄件者E-MAIL)	<input type="text" value="ginger.sina.com"/>
主旨：	<input type="text" value="吳老師你好"/>
內文： (請包含您的聯絡方式及姓名電話)	<input type="text" value="吳老師你好"/>

php 收到這些變數的資料後經過 mail(\$to,\$subject,\$body,\$from)函數的處裏後，就會將資料傳給郵件伺服器(sendmail)，再郵寄出去。

這是我們收到的 email。



1-8-2 瀏覽器上的暫存器 Http Cookies

setcookie() 函數可以將我們上網的資訊存在瀏覽器 (client 端) 上, 帳號與密碼的網頁正是例用這個功能。我們只要將帳號及密碼輸入, 則 setcookie() 函數會將我們的密碼和帳號存在瀏覽器的 Cookie 中, 當網頁要有保密的功能時只要使用 cookie 的帳號和密碼既可驗證, 不用每一次都再輸入帳號和密碼。

如果我們只要讓瀏覽器記錄的資料暫時記憶, 則只要在 setcookie() 函數後面加上時間的參數。

範例 cookie.php

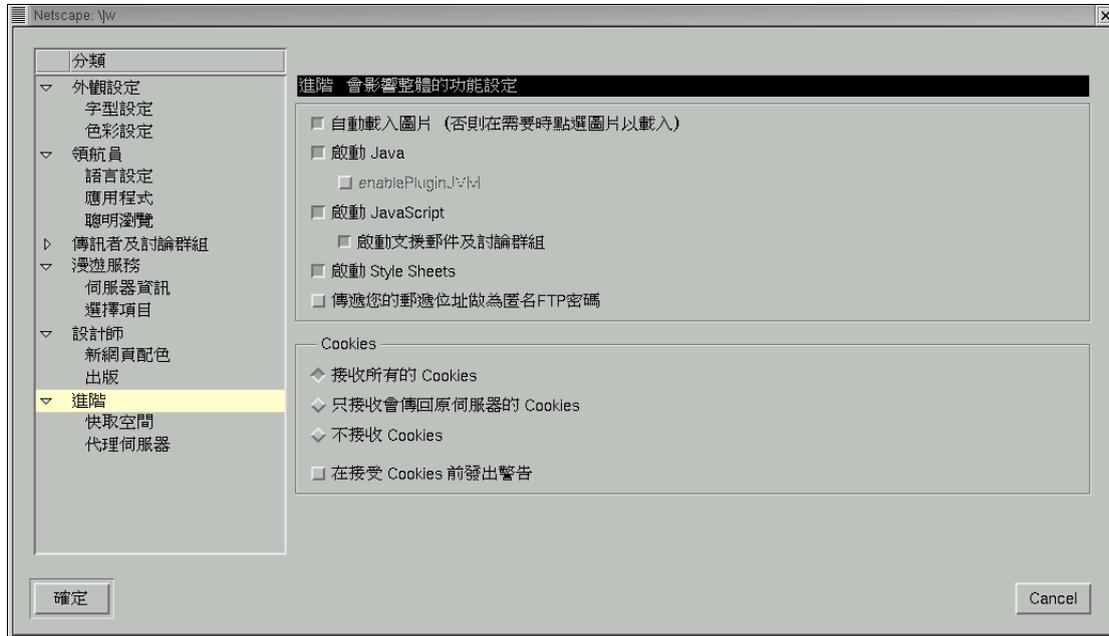
setcookie() 函數將 chaiyen 這個字串存入瀏覽器上並以 login 這個變數來指定, 而瀏覽器只有讓 login 這個變數存在 3600 秒 (一小時), 時間一到就讓它消失。

```
1 <?php
2 setcookie("login", "chaiyen", time()+3600);
3 echo $login;
4 ?>
```

chaiyen 這個字是來自瀏覽器的 cookie 變數 login。



在 Netscape，編輯 -> 偏好設定 -> 進階，就可以打開 Cookies 了。
這是在 Netscape 中 Cookies 的設定，有些網頁因為有密碼驗證所以要打開 Cookies，讓 Cookies 記錄部份資料。



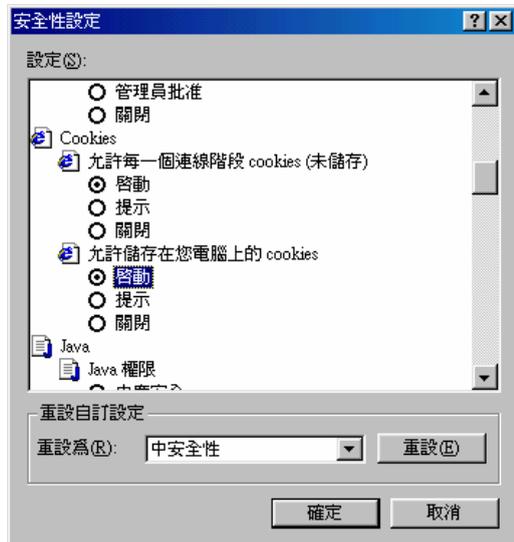
在 IE 瀏覽器上設定 cookie。我們可以先選取工具，Internet 選項。



我們在 Internet 選項選取安全性，我們再選取自訂層級。



在安全性設定我們選取 Cookies，這時就可以設定我們 Cookies 的設定。我們可以選取允許儲存在您電腦上的 cookies。這時有三個層級，我們可以選取啟動。這樣瀏覽器的 Cookies 就可以暫存我們的資料了。



1- 9 常數 constants

我們可以使用 `define()` 函數來定義常數。常數就是不變的數例如在數學上 PI 的值為 3.1415 是不會改變的，在作圖形面積時 PI 的值永遠都是 3.1415 就是常數。有時後為了方便，我們會利用設定常數以方便程式設計的流暢性，例如我們使用 `pi` 來代替 3.1415 時，它的常數值不會改變，就合乎數學的一貫性。

範例 `circle.php`

我們在第二行使用 `Define()` 函數來定義 `pi`。

在第三行直接使用所定義的常數 `pi`。

```
1 <?php
2 define("pi",3.1415);
3 echo 10*10*pi;
4 ?>
```

顯示圓的面積。

