

第十一章 自訂函數

自訂函數



- 除了上述介紹關於日期／時間函數外，我們在前面章節裡介紹了isset()函數判斷表單傳遞的\$_GET或\$_POST系統陣列變數是否存在。
- 如果我們要的功能是系統沒有提供的，或者我們想要把常用的語法集中起來，那就得建立自訂函數。

自訂函數：沒有傳入與傳回值



- 當我們呼叫函數時，沒有將資料傳入函數內，也沒有將資料傳送回去，這樣的函數，該如何規劃呢？

```
function 函數名稱()  
{  
    函數內容;  
}
```

自訂函數：沒有傳入與傳回值

- 例如希望網頁上顯示「網頁內容與圖片均屬版權所有，不得盜用」，而這一段文字若放在函數內，我們可用呼叫函數的方式在網頁上顯示該段文字（檔案名稱：「Phpproject9」資料夾內「function01.php」）。



自訂函數：有傳入沒有傳回值



- 呼叫函數時會將資料傳入函數內，但沒有將資料傳送回去，這樣的函數，該如何規劃呢？

```
function 函數名稱(接收資料的變數)
{
    函數內容;
}
```

自訂函數：有傳入沒有傳回值



- 接收資料的變數於PHP7內可進行型態宣告，這部分請參考11-2-1節介紹。例如依據今天日期的不同，顯示今天是週幾等資訊，那該怎麼做呢？（檔案名稱：「Phpproject9」資料夾內「function02.php」）。



自訂函數：有傳入有傳回值

- 呼叫使用函數時會將資料傳入函數內，但希望函數將計算結果傳送回去，這樣的函數，該如何規劃呢？

```
function 函數名稱(接收資料的變數)
{
    函數內容;
    return 欲傳回的變數;
}
```

自訂函數：有傳入有傳回值



- 傳回的資料於PHP7內可進行型態宣告，這部分請參考11-2-2節介紹。
- 例如我們設計了一個表單，將三個科目成績輸入後，希望告訴我們平均成績，那該怎麼做呢？請先設計表單網頁（檔案名稱：「Phpproject9」資料夾內「function03.html」）。

自訂函數：有傳入有傳回值

- 這個練習限制資料必須為整數型態，如果不在表單進行過濾分析，後端PHP網頁將會接收到非整數型態資料，PHP檢查資料若發現型態不合再返回表單，如此流程反而會讓使用者不便。
- 這個練習裡我們於class1、class2、class3三個單行文字框元件裡偵測鍵盤放開時也就是於onkeyup()事件觸發時執行checknum()函數，而呼叫函數時將執行這個函數的元件當作參數（也就是小括弧裡的this）進行傳遞。

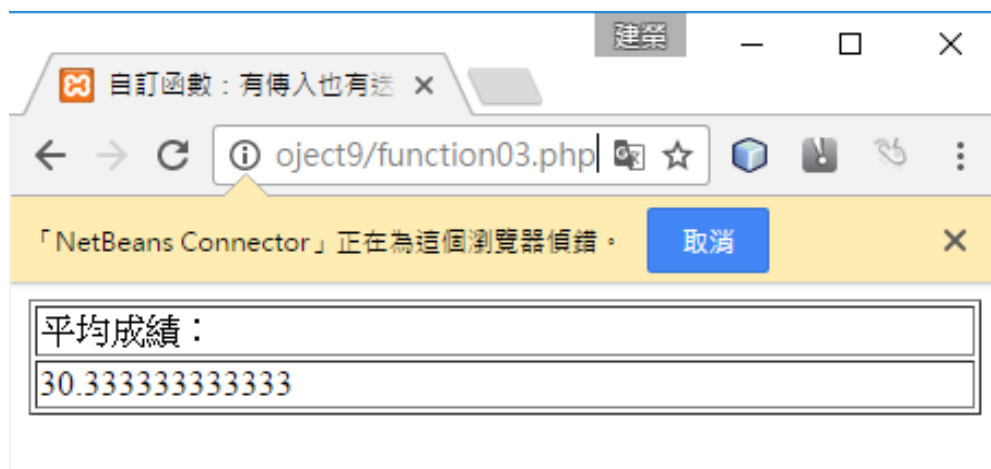
自訂函數：有傳入有傳回值

- `checknum ()`函數將進行資料檢驗，第31行建立一個正規表示式為`re`，`re`測試`obj`是否符合正規表示式，若不符合則彈跳出訊息後將`obj.value`設定為空值，並使`obj`得到滑鼠鍵盤控制權後再傳回`false`。
- 第31行是正規表示式，將該行正規表示式拆解後可依序解釋其用途如下：

驗證語法	說明
<code>^</code>	字串頭
<code>[0-9]</code>	代表 0 至 9 這幾個字元
<code>*</code>	代表 0 或多個
<code>\$</code>	字串尾

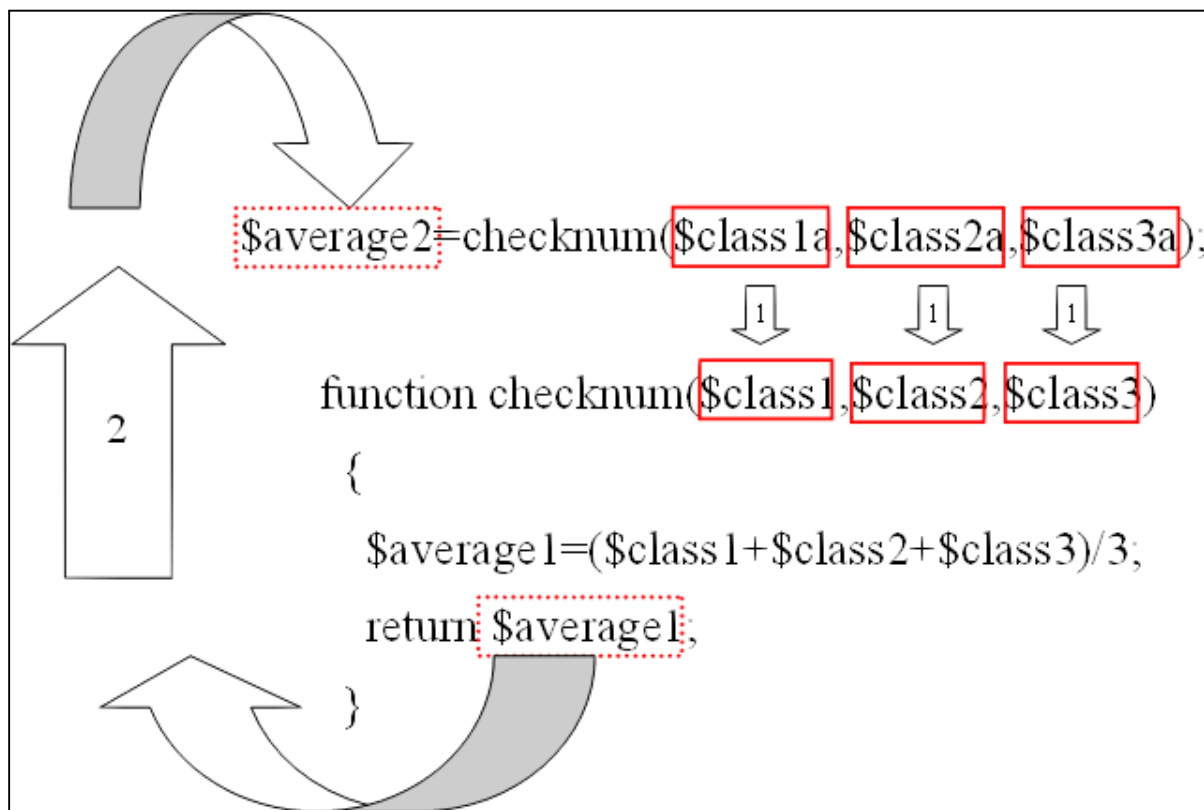
自訂函數：有傳入有傳回值

- 當表單資料送出後，PHP如何處理資料呢（檔案名稱：「Phpproject9」資料夾內「function03.php」）。



自訂函數：有傳入有傳回值

- 自訂函數：有傳入也有送出的執行流程



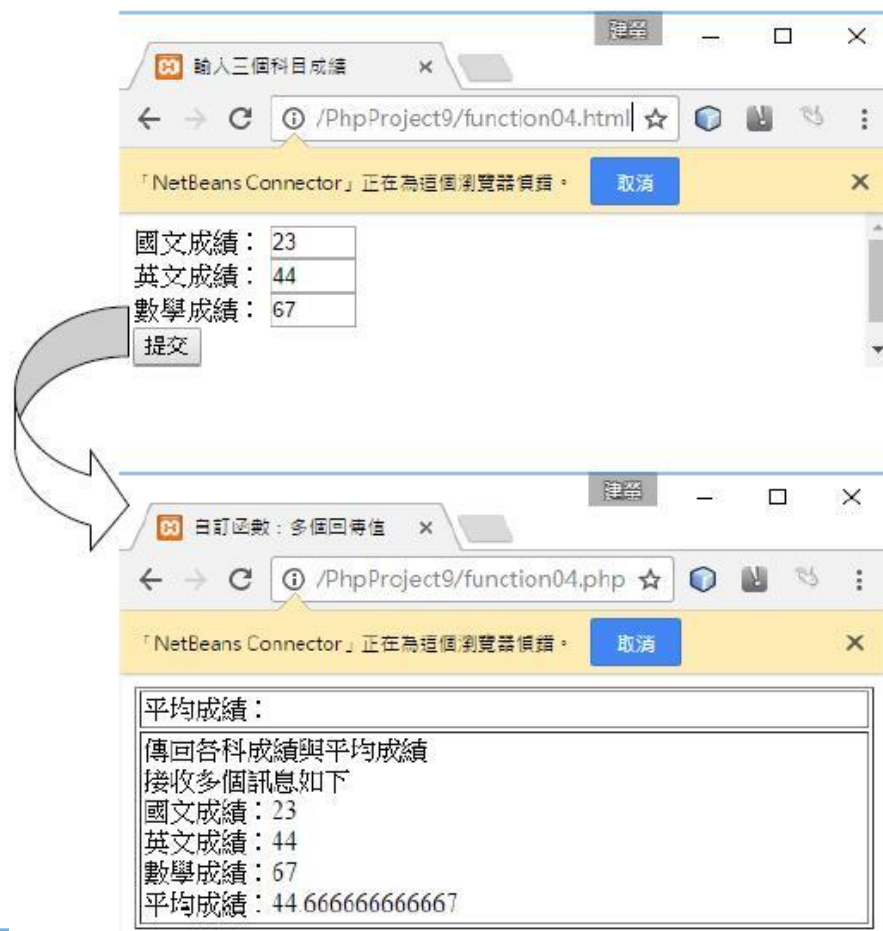
自訂函數：多個回傳值



- 假如函數不止傳回一個變數，而是多個變數，那該怎麼辦呢？
- 以上一個例子為例，假設我們希望函數傳回平均成績、國文成績、英文成績、數學成績，那我該怎麼設計呢？
- 請您將function03.htm另存新檔為function04.htm，並將「action="function03.php"」改為「action="function04.php"」，或請參考檔案名稱：「Phpproject9」資料夾內「function04.htm」，接收網頁可以如此設計（檔案名稱：「Phpproject9」資料夾內「function04.php」）。

自訂函數：多個回傳值

- 自訂函數：多個回傳值。



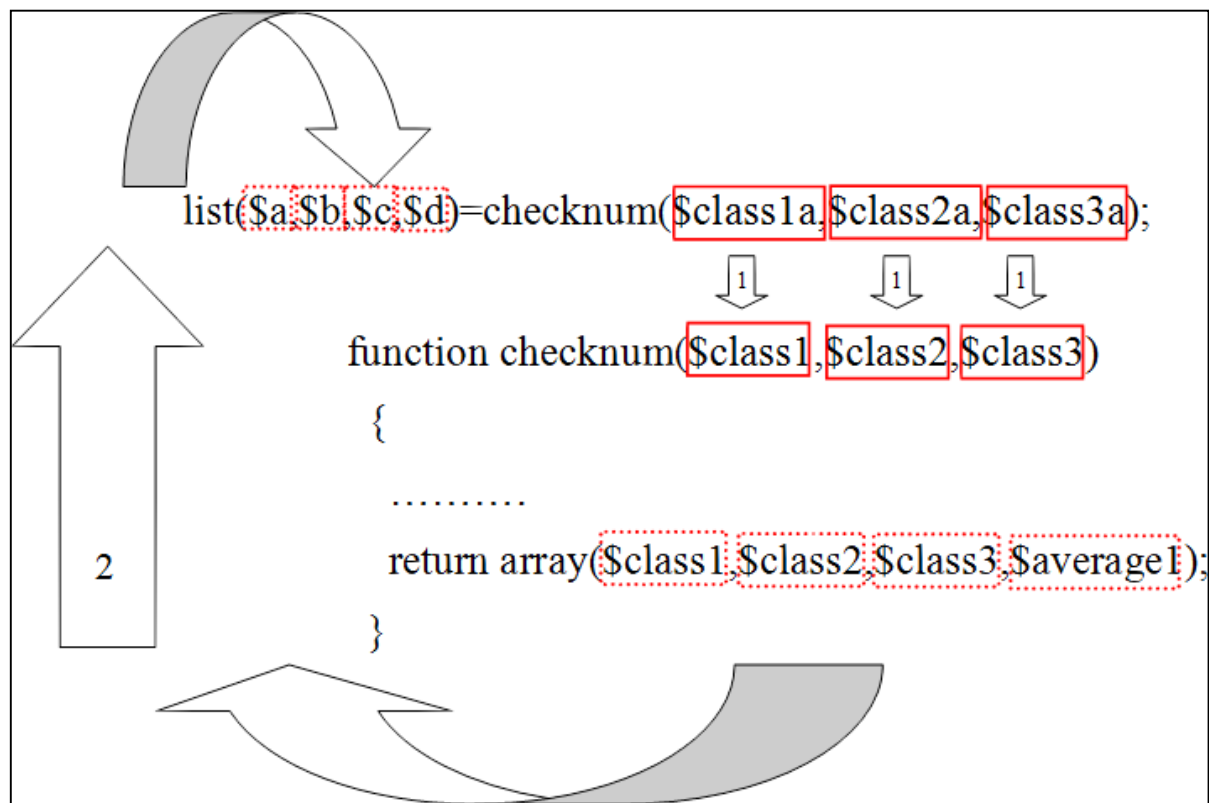
自訂函數：多個回傳值



- 第28行的語法將\$class1a,\$class2a,\$class3a三個變數丟給checknum函數，而第10行checknum函數以\$class1,\$class2,\$class3三個變數接收並處理好資料後在第14行以陣列array(\$class1,\$class2,\$class3,\$average1)變數傳回第18行。
- 第28行以list()函數接收資料，list()函數內設定多個變數，第一個變數\$a對應傳回的\$class1變數，第二個變數\$b對應傳回的\$class2變數，以此類推，接收完資料後就可可在第30行到第33行依序顯示各科成績與平均成績。

自訂函數：多個回傳值

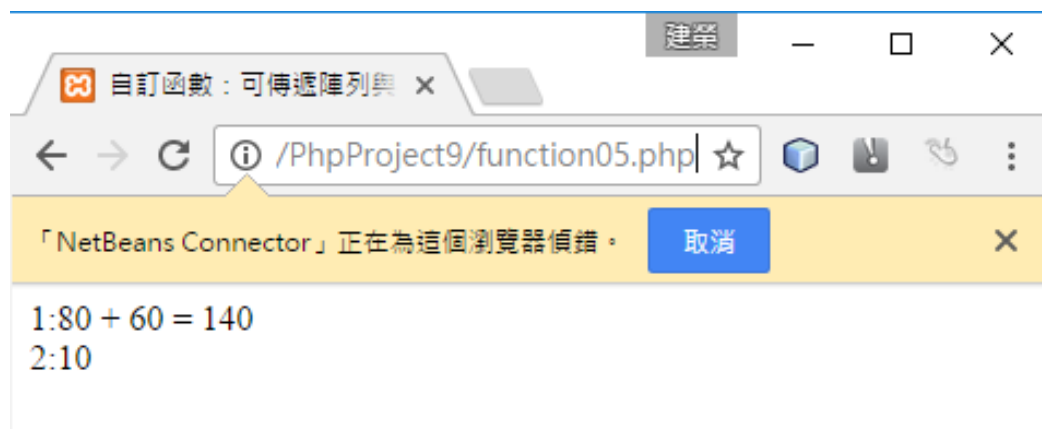
- 自訂函數：多個回傳值的執行流程。



函數接收值進一步設定



- 接收資料時可用陣列收接，接收後再依據陣列索引值逐一進行資料分析處理，另外於PHP5.6以上可以於函數內進行不固定接收數量的參數設定，以「...」方式表示，請參考以下的練習（檔案名稱：「Phpproject9」資料夾內「function05.php」）。



函數接收值進一步設定



- 若您的PHP版本為5.6以下則會以空白方式來處理。
- 第12行傳送一個陣列給第07行的takes_array函數接收，takes_array函數以\$input陣列進行接收，接收後於第09行就依據索引值顯示內容。
- 第13行sum函數接收值是不固定的接收數量，接收後的值都是儲存於\$numbers陣列裡，第15行到第17行則以foreach語法逐一取值後進行加總。

函數接收值進一步設定



- 函數接收值與函數外變數是不同的資料，傳送進去後的變化與外面無關。
- 若希望傳送進去後的資料可以改變原有的內容，接收參數必須加入&符號，請參考以下的練習（檔案名稱：「Phpproject9」資料夾內「function06.php」）。



函數接收值進一步設定

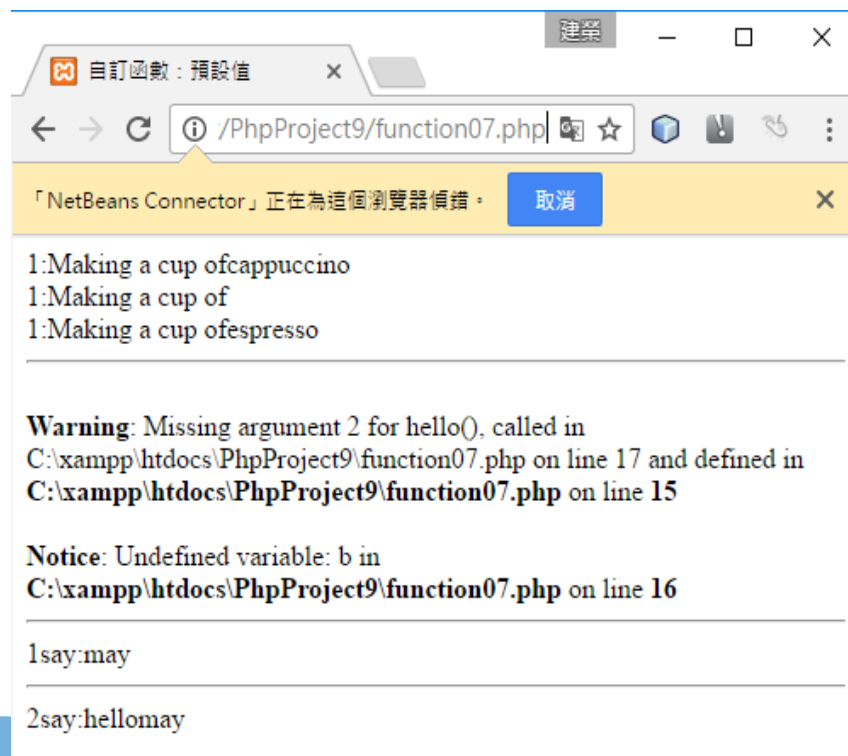


- 兩種不同接收方式：
 - 第07行的add_some1函數以\$string變數接收，函數內與外的資料是不同的。
 - 第16行的add_some2函數以&\$string變數接收，函數內以\$string變數進行操作，函數內與外的資料是相同的。

函數接收值進一步設定



- 函數接收值也可以設定預設值，但如果兩個以上的接收值請記得有預設值的參數請放在最右邊，否則會產生錯誤訊息，請參考以下的練習（檔案名稱：「Phpproject9」資料夾內「function07.php」）。



匿名函數

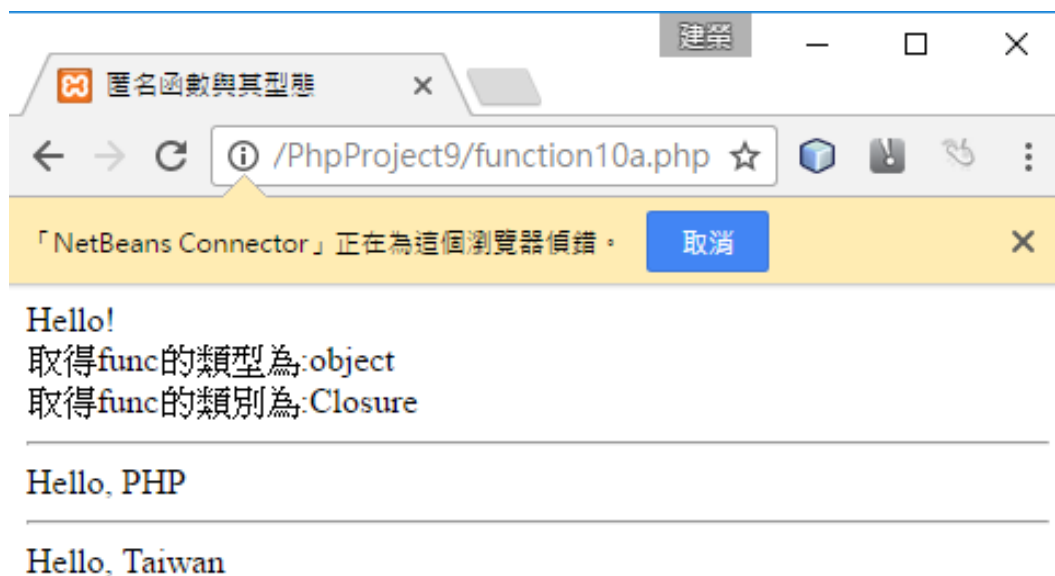


- PHP5.3版以上允許您臨時建立一個沒有指定名稱的函數，這種函數稱之為匿名函數，而這種函數實際上是一個物件，是經由Closure類別產生。
- 操作時您可以把匿名函數當作變數給予內容的方式進行，所以這個函數最後要加上分號。

匿名函數



- 請參考以下的練習，我們將進行三個不同的匿名函數規劃設計（檔案名稱：「Phpproject9」資料夾內「function10a.php」）。



匿名函數



- 您可以直接呼叫匿名函數，第07行到第09行規畫這個匿名函數的結果給\$func，於第12行可以知道\$func它的類型屬於物件，而第14行可以知道\$func它屬於Closure類別，第10行執行\$func物件，執行方式為\$func()，就會呼叫匿名函數。

```
07 $func = function( ) {  
08   echo "Hello!<br>";  
09 };  
10 $func( );  
11 echo "取得 func 的類型為:";  
12 echo gettype($func)."<br>";  
13 echo "取得 func 的類別為:";  
14 echo get_class($func)."<br>";  
15 echo "<hr>";
```


匿名函數



- 您可以傳遞參數到匿名函數內，第16行到第19行規畫這個匿名函數的結果給\$func2，您可看到function之後接收資料，第20行執行\$func2物件，執行方式為\$func2 ('PHP')，就會攜帶資料給匿名函數。

```
16 $func2 = function($name)
17 {
18     echo "Hello, $name<br>";
19 };
20 $func2 ('PHP');
21 echo "<hr>";
```

匿名函數



- 若要從匿名函數外取得變數內容，匿名函數需要使用use語法，第22行規劃\$name變數內容為taiwan，第23行到第25行規畫這個匿名函數的結果給\$func3，您可看到function之後以use接收資料，第26行執行\$func3物件，執行方式為\$func3()，於匿名函數以use(\$name) 攜帶資料給匿名函數。

```
22 $name = 'Taiwan';  
23 $func3 = function( ) use($name) {  
24     echo "Hello, $name<br>";  
25 };  
26 $func3( );
```

匿名函數



- 匿名函數接收外部資料，請您留意前後順序以及函數語法，我們將進行五個不同的匿名函數資料接收練習（檔案名稱：「Phpproject9」資料夾內「function10b.php」）。

A screenshot of a web browser window. The address bar shows the URL "/PhpProject9/function10b.php". Below the address bar, there is a yellow notification bar that says "「NetBeans Connector」正在為這個瀏覽器偵錯。" with a "取消" button. The main content area of the browser displays the output of a PHP script, which consists of six lines of text, each followed by a line number in parentheses:

```
string(5) "hello"  
-1  
string(5) "hello"  
-2  
string(2) "hi"  
-3  
string(2) "hi"  
-4  
string(4) "good"  
-5  
string(5) "world"  
-6  
string(11) "hello world"
```

匿名函數



- 第07行的\$message的內容設定為hello，而自訂函數於第08行進行use動作使用外部的message資料，再於第11行執行時產生的結果是hello。
- 第12行將\$message的內容設定為world，第13行執行第08行的函數語法時，仍依照函數定義輸出hello而非world，而不是呼叫函數時的內容進行傳輸：

```
07 $message = 'hello';  
08 $example = function ( ) use ($message) {  
09     var_dump($message);  
10 };  
11 echo "<br>".$example( )."-1<hr>";  
12 $message = 'world';  
13 echo "<br>".$example( )."-2<hr>";
```

匿名函數



- 第14行將\$message的內容設定為hi，而自訂函數於第15行進行use動作使用外部的message資料，由於函數重新定義呼叫，所以再於第18行執行時產生的結果是hi。
- 第19行將\$message的內容設定為world，第20行執行第15行的函數語法時，這時依照函數定義輸出hi而非world：

```
14 $message = 'hi';  
15 $example = function ( ) use ($message) {  
16     var_dump($message);  
17 };  
18 echo "<br>".$example( )."-3<hr>";  
19 $message = 'world';  
20 echo "<br>".$example( )."-4<hr>";
```

匿名函數



- 第21行將\$message的內容設定為good，而自訂函數於第22行進行use動作使用外部的message資料，由於函數重新定義呼叫，所以再於第25行執行時產生的結果是good，而函數的參數內使用了&符號，代表會指向原本的資料。
- 第26行將\$message的內容設定為world，外部的變數進行資料的改變，第27行執行第22行的函數語法時，這時依照變數的內容輸出world：

```
21 $message = 'good';  
22 $example = function ( ) use (&$message) {  
23     var_dump($message);  
24 };  
25 echo "<br>".$example( )."-5<br>";  
26 $message = 'world';  
27 echo "<br>".$example( )."-6<br>";
```

匿名函數



- 匿名函數可接收參數也可使用外部變數資料，第28行語法可以看到function()內有一個參數\$arg，另外以use方式呼叫\$message變數，所以第31行使用這個函數時有攜帶參數：

```
26 $message = 'world';  
27 echo "<br>".$sexample( )."-6<br>";  
28 $sexample = function ($arg) use ($message) {  
29     var_dump($arg . ' ' . $message);  
30 };  
31 $sexample("hello");
```

php7於函數的新增功能

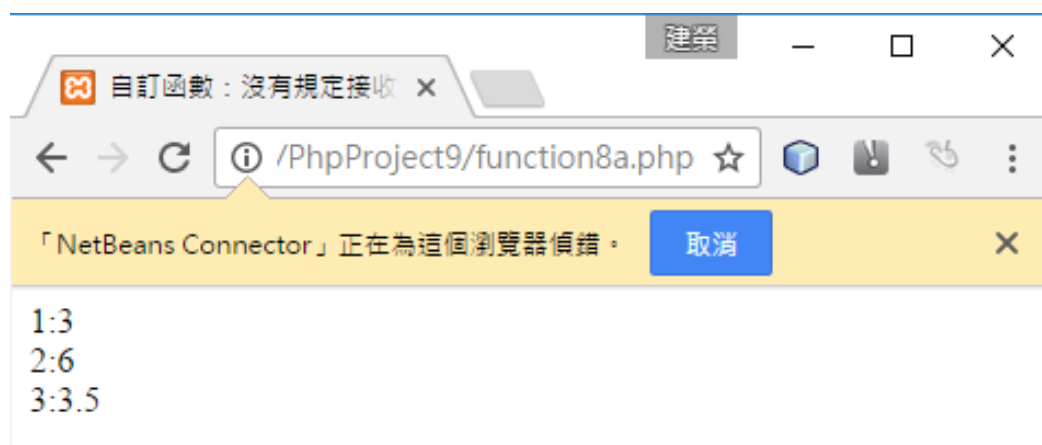


- PHP本身是動態程式語言，變數不需要進行定義，PHP7強化程式安全性，避免不必要的錯誤產生：
 - 接收與傳回資料可分別進行型態定義
 - 可透過「`declare(strict_types=1);`」方式強制規範呼叫函數的網頁文件必須遵守資料型態的定義進行資料的傳送與接收
- 以下各小節說明除非特別提醒，否則於PHP5環境上可能會產生空白畫面的無效操作。

函數接收值型態設定



- 若您沒有規定接收值型態，當我們想要計算總合的結果時可能就會出現一些狀況，例如以下的練習（檔案名稱：「Phpproject9」資料夾內「function8a.php」）。



函數接收值型態設定



- 各種狀況如下：
 - 第10行的1 week就自動轉換為1，變成2加上1成為3。
 - 第11行的3.2自動轉換為浮點數，成為2.8加上3.2，結果為6。
 - 第12行為兩個數值相加，成為3.5。
- PHP7可於函數接收值設定型態，計有int，float，string和bool等型態可以設定，也可以設定為物件或陣列。
- 若接收資料的變數有設定型態，自訂函數的格式改為以下方式呈現：

```
function 函數名稱(型態宣告 接收資料的變數或者物件或者陣列)
{
    函數內容;
}
```

函數接收值型態設定



- 如果函數接收時設定型態，例如設定為int，請查看以下的練習：（檔案名稱：「PhpProject9」資料夾內「function8b.php」）。



函數接收值型態設定

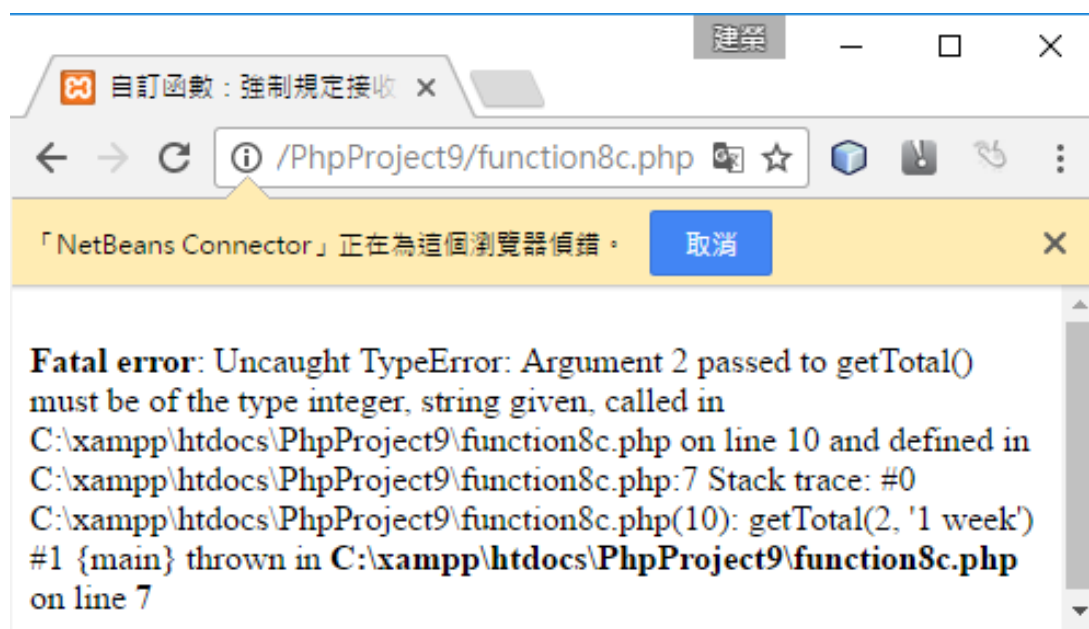


- 各種狀況如下：
 - 第10行的語法不是很洽當，所以系統提出了警告，但1 week仍自動轉換為1，變成2加上1成為3。
 - 第11行的3.2只取出整數值3，2.8也只取出整數值2，成為2加上3，結果為5。
 - 第12行的2.5只取出整數值2再與1相加，成為3。
- PHP7可加入「`declare(strict_types=1);`」語法進行嚴格模式定義，強制規定這個文件的資料必須配合您所宣告的型態，否則將會出現錯誤訊息，請您留意這一行敘述必須放在第一行。

函數接收值型態設定



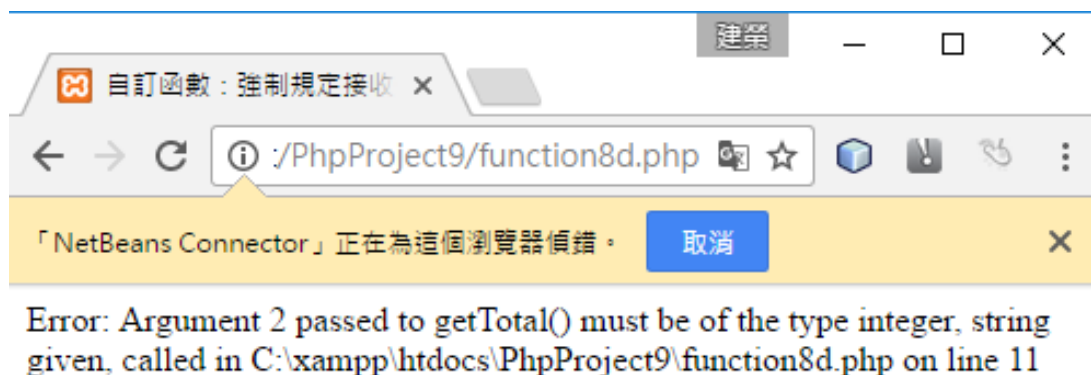
- 如果您的函數是透過呼叫引用的方式載入，請您留意嚴格模式產生效用的範圍是呼叫函數的網頁，而不是定義函數的網頁。請查看以下的練習：（檔案名稱：「Phpproject9」資料夾內「function8c.php」）。



函數接收值型態設定



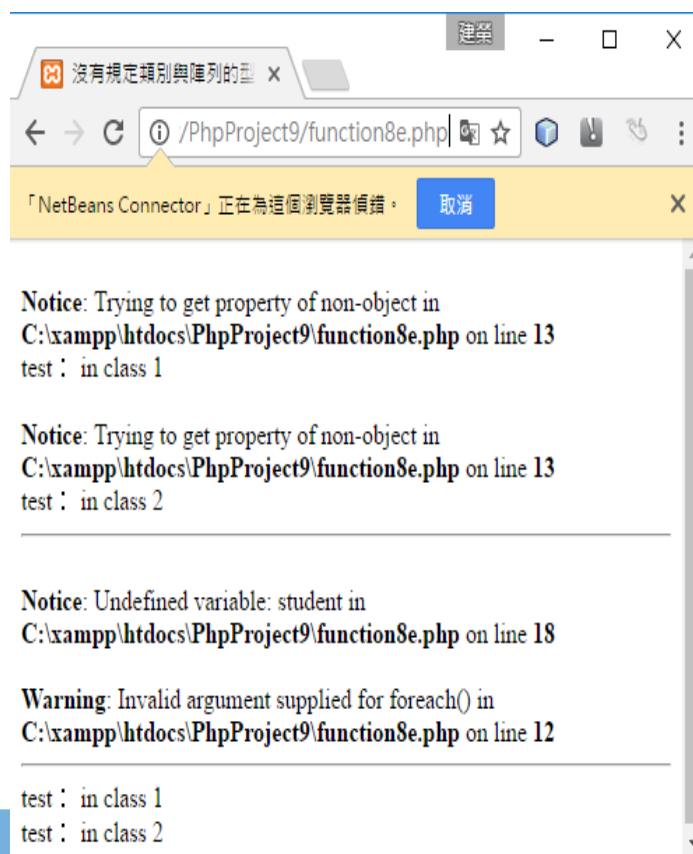
- 加入「`declare(strict_types=1);`」語法進行嚴格模式定義後網頁執行將會產生錯誤，因此建議您可以加入try catch語法進行例外的處理，請查看以下的練習：（檔案名稱：「Phpproject9」資料夾內「function8d.php」）。



函數接收值型態設定



- 如果函數接收的資料是物件與陣列，那接收時一定要做型態的宣告，不然程式會產生問題，請查看以下的練習：
（檔案名稱：「Phpproject9」資料夾內
「function8e.php」）。



函數接收值型態設定

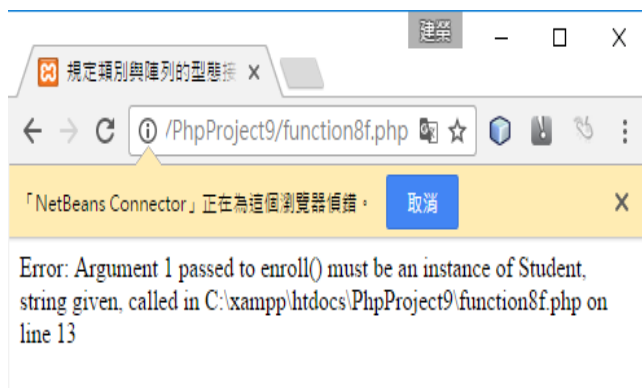


- function8e.php執行問題：
 - 第16行因“name”是字串而非物件，所以產生錯誤訊息。
 - 第18行則因\$student並未定義為物件，而"class"不是陣列而產生問題。
- 若要接收物件與陣列，函數的接收值請宣告資料型態。

函數接收值型態設定



- 由於function8e.php的程式碼執行時會有問題，所以我們在此加入try catch語法進行例外的處理，請查看以下的練習：
（檔案名稱：「Phpproject9」資料夾內「function8f.php」）。
- 自訂函數：顯示第13行例外狀況
- 自訂函數：第13行設定註解後，顯示第15行例外狀況



返回資料型態設定



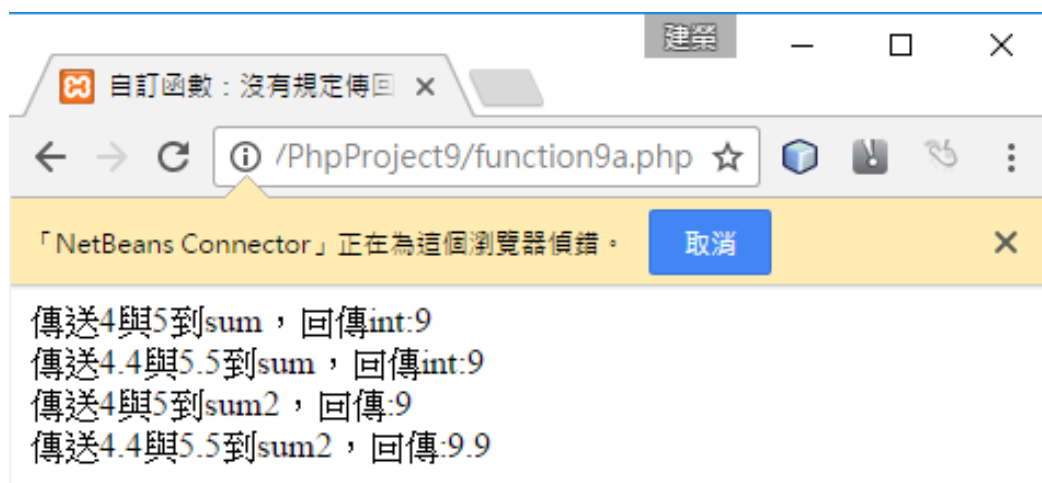
- 函數除了接收值可以進行型態的定義外，函數的傳回值也可以進行型態定義，接收與傳回這兩部分不一定同時都要設定型態。
- 若傳回值設定型態，自訂函數的格式改為以下方式呈現：

```
function 函數名稱(型態宣告 接收資料的變數或者物件或著陣列):資料型態  
{  
    函數內容;  
    return 欲傳回的變數;  
}
```

返回資料型態設定



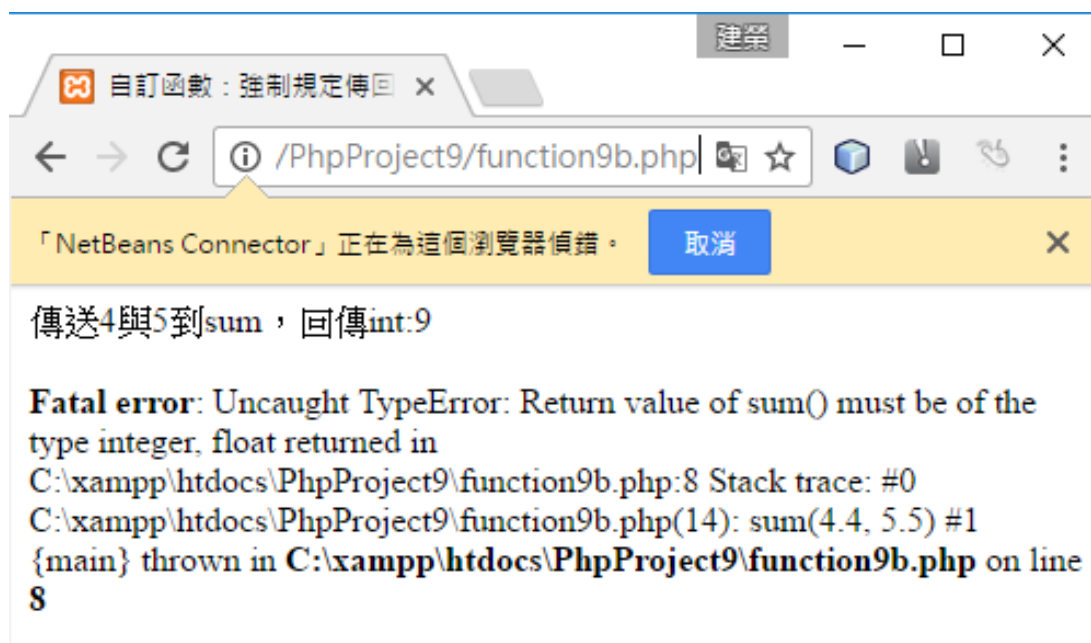
- 函數除了接收值可以進行型態的定義外，函數的傳回值也可以進行型態定義，請參考以下的練習（檔案名稱：「Phpproject9」資料夾內「function9a.php」）。



返回資料型態設定



- 嚴格模式「`declare(strict_types=1);`」語法也會規範傳回值函數的操作，所以相同的練習若在第1行加入嚴格模式，請參考以下的練習（檔案名稱：「Phpproject9」資料夾內「function9b.php」）。



讀取保護私有成員之各種做法



- PHP是一個動態程式語言，對於權限的要求不像C++/Java般嚴謹，主要仍要依賴程式設計師養成良好的習慣。
- 如果現在必須存取類別內的保護或著私有成員，那該如何進行呢？從PHP5開始提供多種方式讓您進行保護或著私有成員的存取。

讀取保護私有成員之各種做法



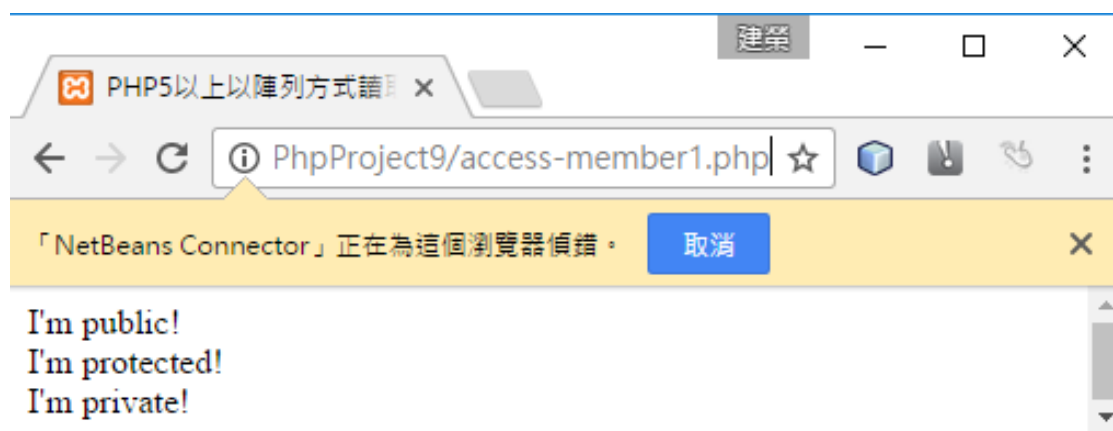
- 以下的各節練習前面14行語法相同，均是產生一個名為 \$obj 的物件，所以以下各節就省略這14行語法：

```
01 <!DOCTYPE html>
02 <html><head>
03   <meta charset="UTF-8">
04   <title>讀取類別內保護與私有欄位</title>
05 </head>
06 <body>
07   <?php
08 class MyExampleClass
09 {
10     public $myPublicVar = "I'm public!";
11     protected $myProtectedVar = "I'm protected!";
12     private $myPrivateVar = "I'm private!";
13 }
14 $obj = new MyExampleClass();
```

以陣列方式讀取



- PHP5可以使用陣列的方式讀取物件內的保護私有成員，這裡僅列出15行之後的語法，之前的語法請參考11-3節說明（檔案名稱：「Phpproject9」資料夾內「access-member1.php」）。



以陣列方式讀取



- 第15行將物件轉換為陣列，接著做不同存取方式：
 - 第16行讀取公開權限資料。
 - 第17行以["\0*\0". '保護等級成員名稱']方式讀取保護等級成員內容。
 - 第18行以["\0". '類別名稱'. "\0". '私有等級成員名稱']的方式讀取私有等級成員內容。

物件導向處理



- PHP5.3版新增物件的處理方式，您可以利用ReflectionObject方法建立一個映像物件，利用這個映像物件進行資料的存取。
- 這裡僅列出15行之後的語法，之前的語法請參考11-3節說明（檔案名稱：「Phpproject9」資料夾內「access-member2.php」）。

A screenshot of a web browser window. The address bar shows the URL "/PhpProject9/access-member2.php". Below the address bar, there is a yellow notification bar that says "「NetBeans Connector」正在為這個瀏覽器偵錯。" with a "取消" (Cancel) button. The main content area of the browser displays the text "I'm protected!" and "I'm private!" on separate lines. At the bottom, a red error message is visible: "Fatal error: Uncaught ReflectionException: Cannot access non-public member MyExampleClass::myPrivateVar in C:\xampp\htdocs\PhpProject9\access-member2.php:23 Stack trace: #0 C:\xampp\htdocs\PhpProject9\access-member2.php(23): ReflectionProperty->getValue(Object(MyExampleClass)) #1 {main} thrown in C:\xampp\htdocs\PhpProject9\access-member2.php on line 23".

```
Fatal error: Uncaught ReflectionException: Cannot access non-public member
MyExampleClass::myPrivateVar in C:\xampp\htdocs\PhpProject9\access-
member2.php:23 Stack trace: #0 C:\xampp\htdocs\PhpProject9\access-
member2.php(23): ReflectionProperty->getValue(Object(MyExampleClass)) #1
{main} thrown in C:\xampp\htdocs\PhpProject9\access-member2.php on line
23
```

物件導向處理



- ReflectionObject方法建立映像物件方法太耗損資料，所以PHP5.4版起新增匿名函數，而匿名函數所屬的Closure類別可以進行成員的繫結，所以可以利用這樣的特性讀取資料。
- 這裡僅列出15行之後的語法，之前的語法請參考11-3節說明（檔案名稱：「Phpproject9」資料夾內「access-member3.php」）。

新增call方法



- PHP7於存取保護私有成員這部分提供了call方法，這裡僅列出15行之後的語法，之前的語法請參考11-3節說明（檔案名稱：「Phpproject9」資料夾內「access-member4.php」）。
- 第15行產生匿名函數，第16行呼叫物件的保護等級成員，這樣的操作方式是不是快速很多？

問題與討論



- 1.請輸入身高體重，計算是否符合BMI標準體重範圍，而BMI標準體重範圍是一個自訂函數。
- 2.請設計一個網頁，讓使用者輸入12個月的溫度，可計算年平均溫度、由大而小排序或由小而大排序，以上三種計算方式請以自訂函數方式處理。
- 3.請實作一個案例解釋匿名函數。
- 4.請設計一個網頁，可以進行自訂函數的指定型態資料的接收。

問題與討論



- 5.請設計一個網頁，可以進行自訂函數的指定型態資料的傳回。
- 6.「`declare(strict_types=1);`」代表甚麼意思？
- 7.請教PHP如何讀取類別裡的保護與私有成員？請實作一個案例解釋。