



第十章 例外處理與日期時間資料

例外處理



- 所謂例外(Exception)是指程式發生不正常的錯誤，進而導致無法繼續執行的情形。
- 例外處理(Exception Handling)就是例外發生時的處理方式。
- 您可能認為說我們就做if條件分析就可以排除狀況，可是您很難捕捉到物件執行操作時何時何處何種狀況下產生狀況，所以傳統的if條件分析只能針對已知的狀況進行分析。

例外處理



- 例外處理語法架構為try catch，寫法如下：

```
try
{
    程式執行中斷的兩種條件：
    1.執行時產生 PHP 已知的錯誤或例外則停止執行後拋至 catch 處理例外。
    2.執行時符合我們設定的條件，停止執行後拋至 catch 處理例外。
}
catch (例外類別) {
    產生例外時的處理區塊。
}
```

例外處理

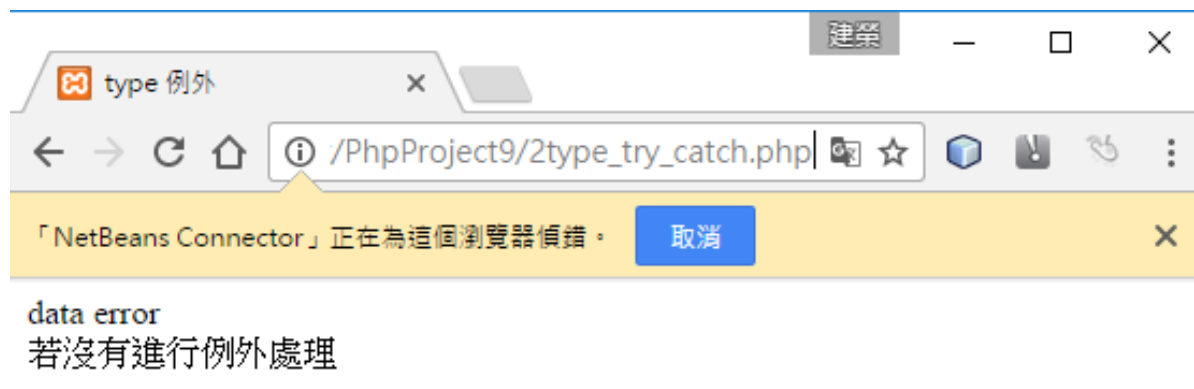


- 依照PHP7的類別架構，Error 類別並不是從 Exception類別擴展出來，所以處理錯誤時不可以使用 `catch (Exception $e) { ... }` 方式，而必須採用 `catch (Error $e) { ... }` 的方式才可以處理。
- 若要細分Error類別內的項目，目前PHP7提供了 ArithmeticError、AssertionError、DivisionByZeroError、ParseError與TypeError等五種Error類別，筆者建議若不清楚Error所屬類別，您可以使用Error進行例外捕捉就可以了。

Error捕捉



- 當我們想要進行兩個數字的加總，可是如果使用使用者帶進去的資料是字串，那該怎麼辦呢（檔案名稱：「Phpproject9」資料夾內「2type_try_catch.php」）。

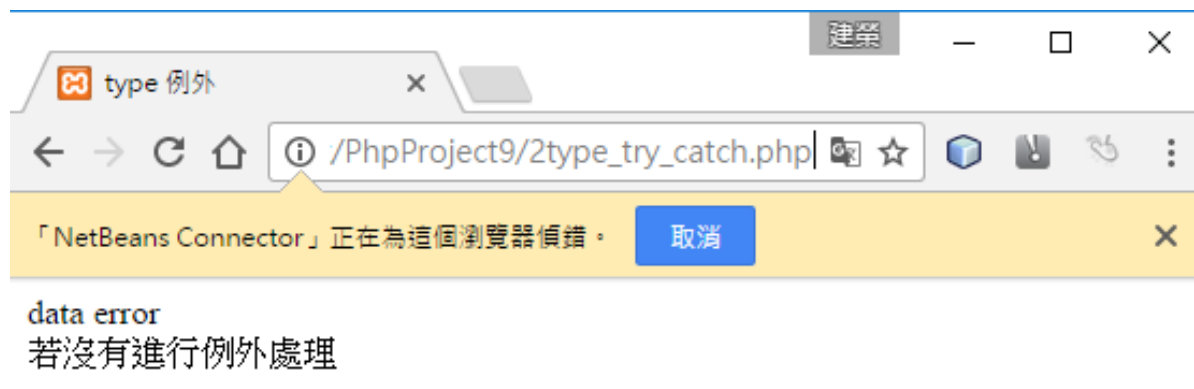


Fatal error: Uncaught TypeError: Argument 1 passed to add() must be of the type integer, string given, called in C:\xampp\htdocs\PhpProject9\2type_try_catch.php on line 16 and defined in C:\xampp\htdocs\PhpProject9\2type_try_catch.php:7 Stack trace: #0 C:\xampp\htdocs\PhpProject9\2type_try_catch.php(16): add('left', 'right') #1 {main} thrown in C:\xampp\htdocs\PhpProject9\2type_try_catch.php on line 7

Error捕捉



- 當我們想要進行兩個數字的加總，可是如果使用使用者帶進去的資料是字串，那該怎麼辦呢（檔案名稱：「Phpproject9」資料夾內「2type_try_catch.php」）。

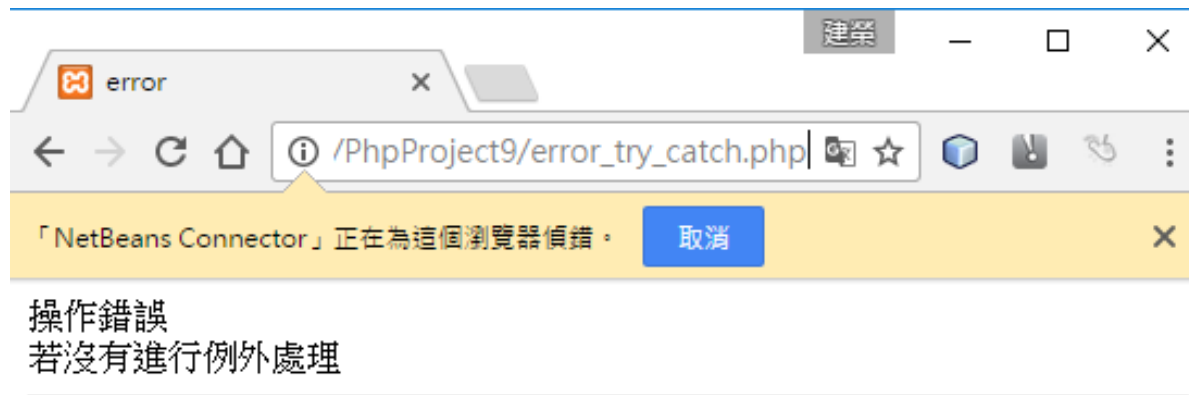


Fatal error: Uncaught TypeError: Argument 1 passed to add() must be of the type integer, string given, called in C:\xampp\htdocs\PhpProject9\2type_try_catch.php on line 16 and defined in C:\xampp\htdocs\PhpProject9\2type_try_catch.php:7 Stack trace: #0 C:\xampp\htdocs\PhpProject9\2type_try_catch.php(16): add('left', 'right') #1 {main} thrown in C:\xampp\htdocs\PhpProject9\2type_try_catch.php on line 7

Error捕捉



- 透過TypeError類別我們可以捕捉到型態錯誤這樣的例外情形，`$e->getMessage()`代表捕捉到的系統例外訊息。
- 如果我們要進行的動作是不存在的，那系統該如何反映呢（檔案名稱：「Phpproject9」資料夾內「error_try_catch.php」）。

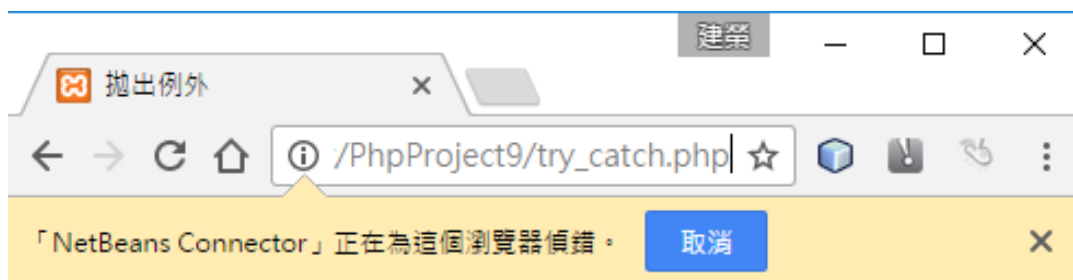


Fatal error: Uncaught Error: Call to a member function method() on integer in C:\xampp\htdocs\PhpProject9\error_try_catch.php:15 Stack trace: #0 {main} thrown in C:\xampp\htdocs\PhpProject9\error_try_catch.php on line 15

手動拋出例外



- 有些狀況並不是系統所預期的，而是我們認為這樣的資料不應該出現，這時就得加入條件判斷式以及手動拋出的動作。
- 例如每一個月最多是31天，若大於31則代表例外得拋出例外（檔案名稱：「Phpproject9」資料夾內「try_catch.php」）。



日期不對:32

時區調整與格式化參數



- PHP網頁時區：
 - PHP4時代不需設定
 - PHP 5若沒有作設定調整則預設為格林威治時間（與臺灣差八小時）
 - PHP7預設您一定要在PHP網頁內設定時區
- PHP官方網站上提供了時區資訊：
<http://www.php.net/manual/en/timezones.php>
- PHP提供的時區分類，共有美洲、亞洲、歐洲等多個群組，在亞洲群組內，您可看到各種不同時區，台灣請挑選「Asia/Taipei」。

php.ini內時區設定調整



- 如果您使用的是Windows作業系統，請您搜尋php.ini這個檔案，開啟檔案後請點選「編輯」功能表內的「尋找」，請尋找「timezone」。
- 如果您使用的是Linux系列作業系統，請您使用「find / -name 'php.ini」的方式搜尋，若無特殊設定，一般可在/etc目錄內找到。若找到的檔案是「/etc/php.ini」，請先切換到root（請下指定：su）身份後輸入「vi /etc/php.ini」。再進入php.ini編輯畫面後，請輸入「/」後再請您輸入「timezone」後按下enter。

php.ini內時區設定調整



- 您會找到以下的字串：

[Date]

; Defines the default timezone used by the date functions

;date.timezone =

php.ini內時區設定調整



- 請您將;date.timezone =前面的;刪除後，在=後面加上時區，例如若您若設定為台北時間，您應該改為：

[Date]

; Defines the default timezone used by the date functions

date.timezone = Asia/Taipei

語法上時區設定調整



- PHP提供的傳統語法上ini_set()函數設定時區以及date()與mktime()兩個函數設定日期時間格式化輸出。
- 但傳統語法設定會碰到2038年問題，所以日期時間有關的語法建議您開始調整到物件導向語法設計。
- 請依據DateTime 類別建立一個DateTime 型態的物件，而DateTime 類別的建構子內有兩個參數：
 - 第一個是日期時間字串
 - 第二個是時區

語法上時區設定調整



- 時區請依據DateTimeZone類別輸入時區名稱建立物件，例如輸入Asia/Taipei代表建立亞洲台北時區。
- 若您要規劃設定為台北時間，在此建立一個物件\$dateTime，物件導向語法上時區語法為：

```
$dateTime = new DateTime("now", new DateTimeZone('Asia/Taipei'));
```

2038年問題

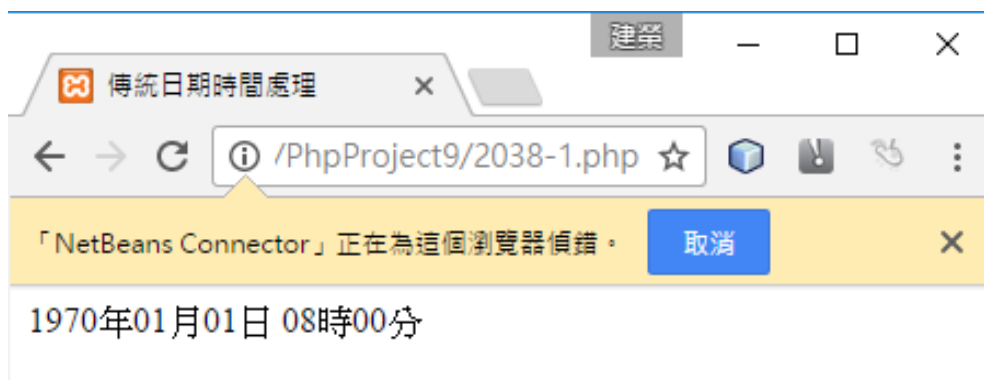


- PHP以4個位元組儲存日期時間資料，以1970年1月1日凌晨0時0分0秒作為時間的起點（這個時間點的值為0），從這裡開始的時間是一秒一秒累積計算。
- 如果時間已經累積到了1349879415這個數值，代表這時距離1970年1月1日凌晨0時0分0秒已經過去了1349879415秒，換算一下就是2012年10月10日14時30分15秒。4個位元組儲存空間最大值為2147483647，所以當時間跳完2147483647秒，值就會轉為負數，也就是時間將會變成無效。

2038年問題



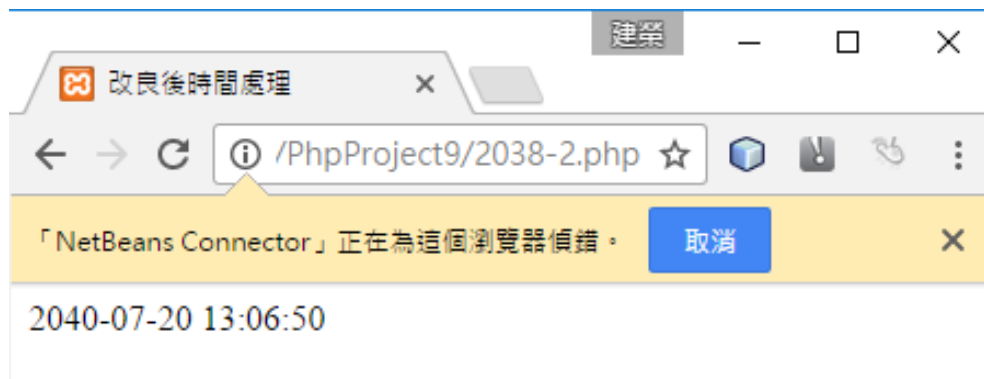
- 而2147483647秒這個時間是2038年1月19日03:14:07，過了這個時間點現有的系統時間都會產生意外。
- 傳統語法上日期時間處理如7-1-3節所介紹的會有2038年的問題，例如我們以傳統語法方式顯示2040年7月20日的年月日時分秒資料（檔案名稱：「Phpproject9」資料夾內「2038-1.php」）。



2038年問題



- 目前PHP提供的傳統語法上時區設定與日期時間格式化輸出均會碰到2038年問題，您必須逐漸修改網頁上的日期時間相關語法，改以物件導向語法方式規劃時區設定與日期時間格式化輸出，就可以避免2038年問題，請參考以下修改的範例（檔案名稱：「Phpproject9」資料夾內「2038-2.php」）。



日期時間字串格式化



- 您必須使用字串格式化方式設定好您想輸出的日期或時間格式。
- 首先介紹與日期有關的格式化參數，請注意大小寫代表的意義不同：

參數	說明
d	小寫 d，代表日期，以數字表示，例如："01" 到 "31"
j	小寫 j，代表日期，以數字表示，但是不足 2 位數不補 0，例如："1" 到 "31"
D	大寫 D，代表星期幾，以 3 個英文字表示，例如："Sun"
l	小寫 L，代表星期幾，以英文全名表示，例如："Sunday"
w	小寫 w，以數字表示星期幾，例如："0" 到 "6"
m	小寫 m，代表幾月，例如："01" 到 "12"
n	小寫 n，代表幾月，不足 2 位數不補 0，例如："1" 到 "12"
M	大寫 M，代表幾月，以 3 個英文字表示，例如："Oct"
F	大寫 F，代表幾月，以英文全名表示，例如："October"
Y	大寫 Y，代表西元幾年，以 4 位數表示，例如："1999"
y	小寫 y，代表西元幾年，以 2 位數表示，例如："99"
t	小寫 t，代表當月的天數，例如："28" 到 "31"
z	小寫 z，代表一年中的第幾天，例如："0" 到 "365"
L	大寫 L，判斷是否為閏年
W	大寫 W，今天是一年中的第幾週。

日期時間字串格式化



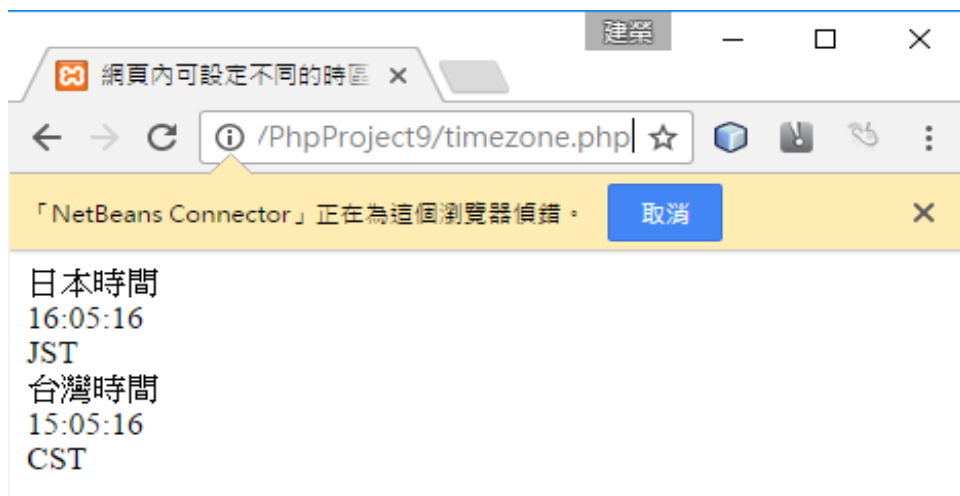
- 接著介紹與時間有關的格式化參數，請注意大小寫代表的意義不同：

參數	說明
h	小寫 h，代表小時，12 小時制，例如：" 01 " 到 " 12 "
g	小寫 g，代表小時，12 小時制不足 2 位數不補 0，例如：" 1 " 到 " 12 "
H	大寫 H，代表小時，24 小時制，例如：" 00 " 到 " 23 "
G	大寫 G，代表小時，24 小時制不足 2 位數不補 0，例如：" 0 " 到 " 23 "
a	小寫 a，"am" 或 "pm"
A	大寫 A，"AM" 或 "PM"
i	小寫 i，代表幾分，例如：" 00 " 到 " 59 "
s	小寫 s，代表幾秒，例如：" 01 " 到 " 59 "
T	大寫 T，代表目前設定的時區

日期/時間格式化輸出



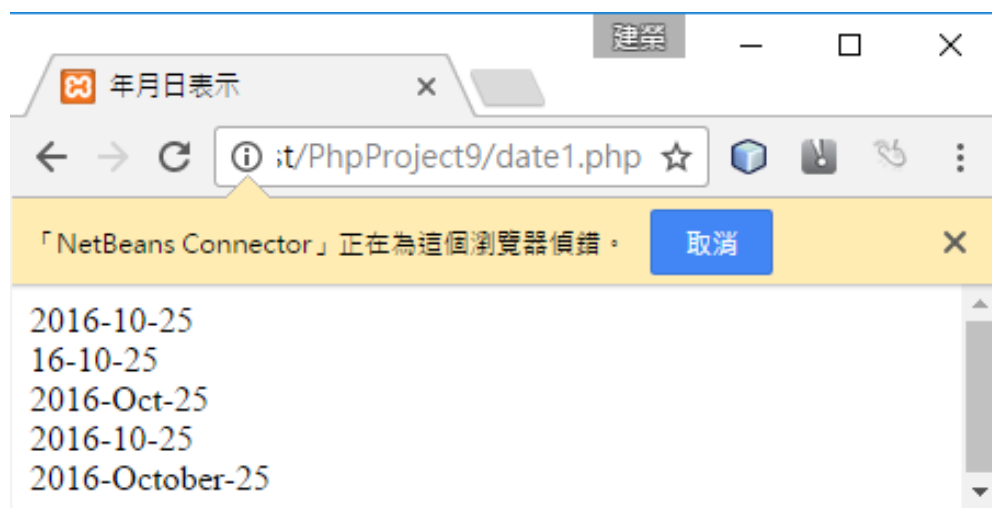
- 日期時間格式化輸出是指「將日期或時間依照我們的要求顯示」，依據DateTime 類別建立的物件可藉由執行format() 方法的方式進行格式化的輸出。
- 請於網頁上加入兩個時區設定，且於format()方法內加入年月日時分秒等各種日期時間格式化參數（檔案名稱：「Phpproject9」資料夾內「timezone.php」）。



日期/時間格式化輸出



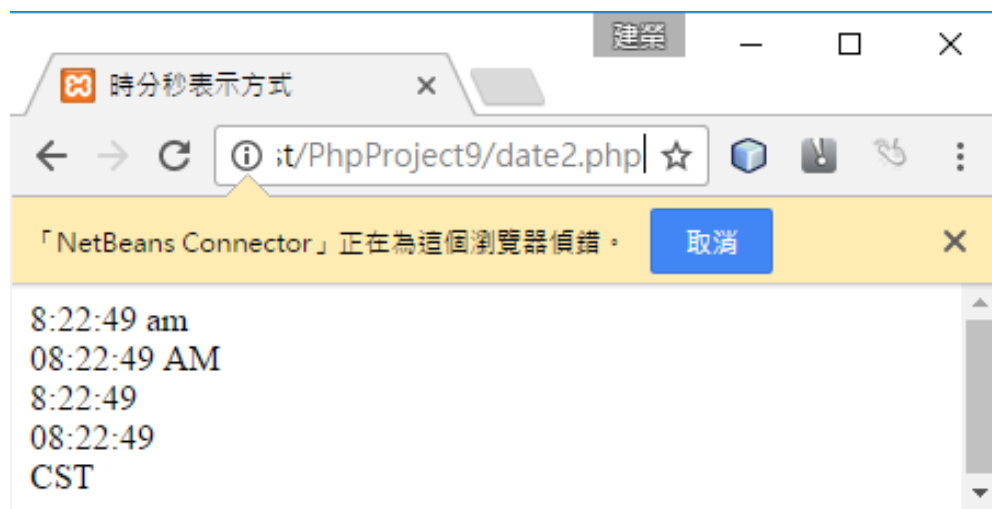
- 我們可將多個與日期有關的參數加入日期時間類別物件的 `format()` 方法內運用，請於日期時間類別物件的 `format()` 方法內加入年月日等各種日期格式化參數（檔案名稱：「Phpproject9」資料夾內「date1.php」）。



日期/時間格式化輸出



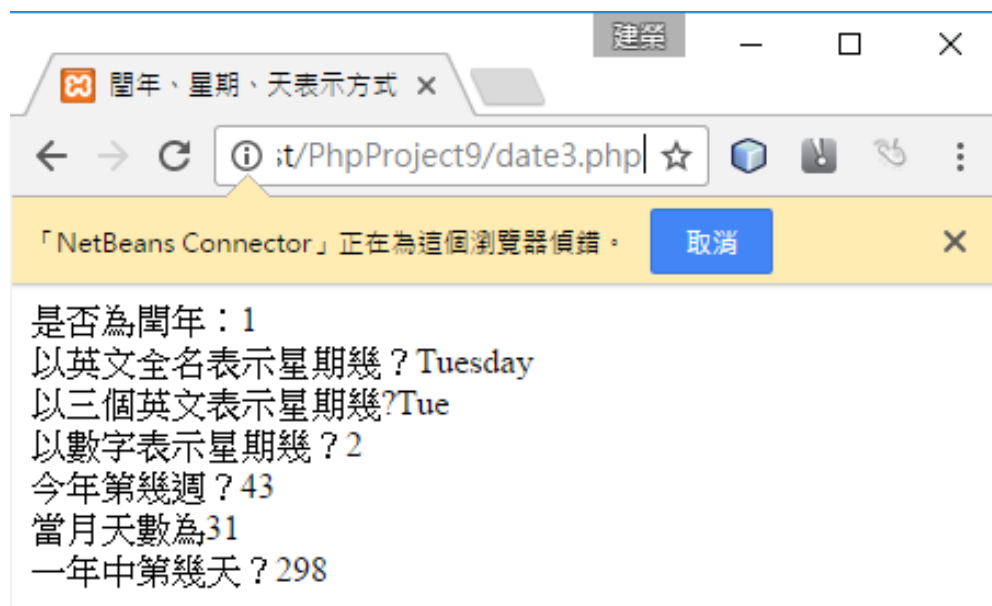
- 我們可將多個與時間有關的參數加入日期時間類別物件的 `format()` 方法內運用，請於日期時間類別物件的 `format()` 方法內加入時分秒等各種時間格式化參數（檔案名稱：「Phpproject9」資料夾內「date2.php」）。



日期/時間格式化輸出



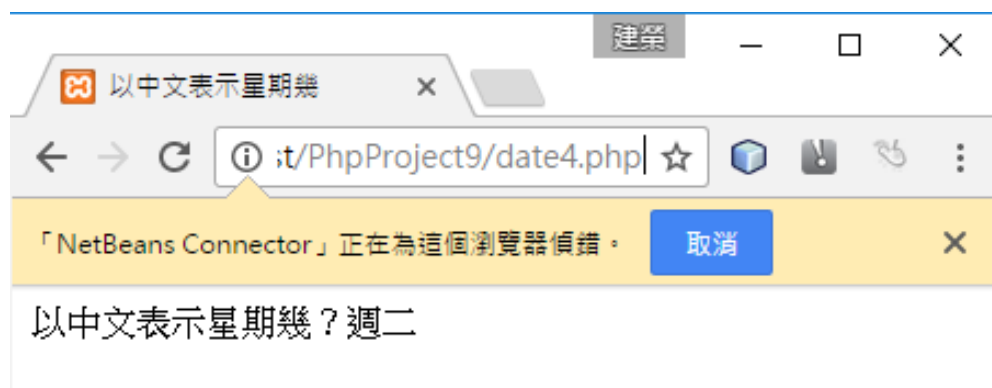
- 我們可將多個與閏年、星期與天有關的參數加入日期時間類別物件的format()方法內運用，請於日期時間類別物件的format()方法內加入閏年、星期與天等各種日期時間格式化參數（檔案名稱：「Phpproject9」資料夾內「date3.php」）。



日期/時間格式化輸出



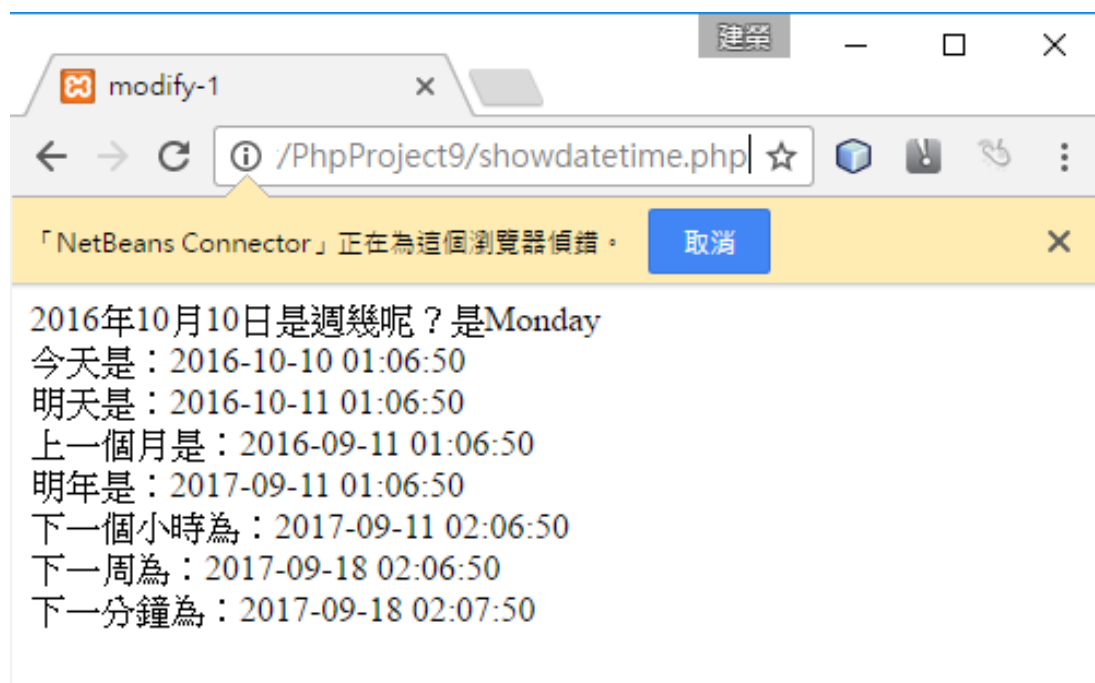
- 系統所提供的顯示星期幾的參數只可以顯示英文的星期資訊，如果要顯示中文的星期資訊，可以使用很多種方式。
DateTime是一個類別，所以我們可以自訂一個類別，透過這個自訂的類別顯示中文的星期資訊，請參考以下的範例（檔案名稱：「Phpproject9」資料夾內「date4.php」）。



設定日期時間與調整



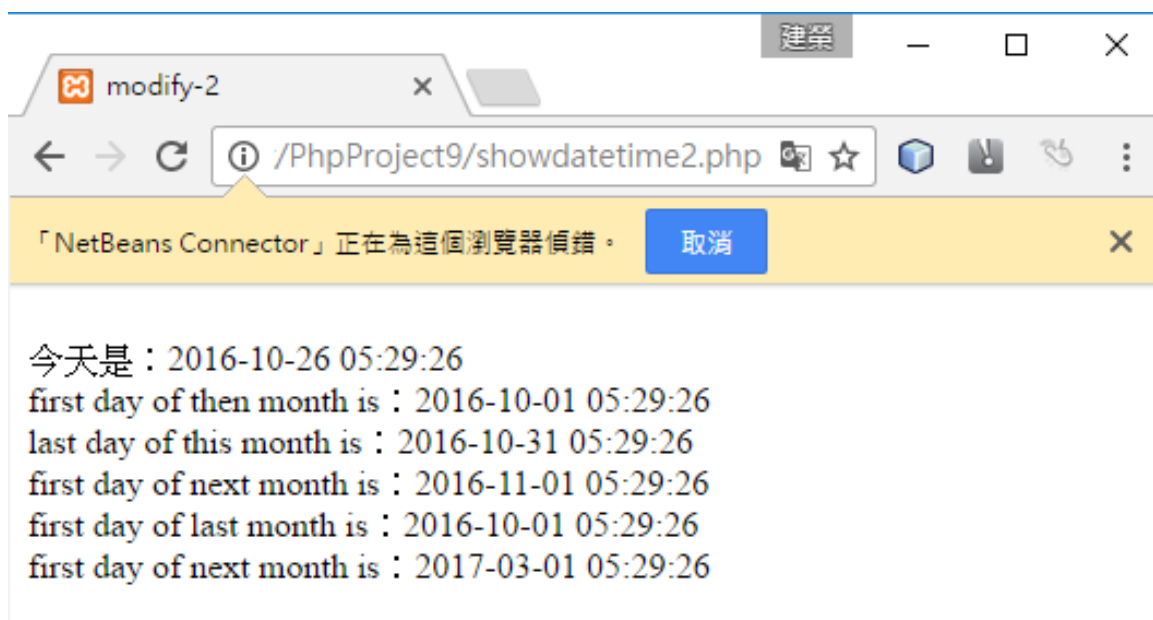
- 請設計一個網頁，顯示固定的日期、明天、上一個月、明年、下一個小時、下一周與下一分鐘的年月日時分秒資訊，請進行設計（檔案名稱：「Phpproject9」資料夾內「showdatetime.php」）。



設定日期時間與調整



- 我們也可以在DateTime 類別建立的物件之modify()方法內加入first、last、this、next等詞彙調整為特定的日期，請進行設計（檔案名稱：「Phpproject9」資料夾內「showdatetime2.php」）。



字串轉日期時間型態資料



- 我們可以從表單網頁輸入日期時間資料，表單輸入的元件可以挑選text或著date，date是html5新增的表單元件（檔案名稱：「Phpproject9」資料夾內「strtotime1.html」）。

A screenshot of a web browser window displaying a form for date and time input. The browser's address bar shows the URL "localhost/PhpProject9/strtotime1.htm". A yellow notification bar at the top states "「NetBeans Connector」正在為這個瀏覽器偵錯。" with a "取消" (Cancel) button. The form contains the following elements:

- Text: "可以比較text與date的差別"
- Text: "以text型態輸入"
- Text input: "輸入第1段時間:" with the value "test"
- Text input: "輸入第2段時間:" with the value "20161026"
- Text: "以date型態輸入"
- Text input: "輸入第3段時間:" with the value "2016/10/26"
- Text input: "輸入第4段時間:" with the value "2016/10/29" and a dropdown arrow.
- Buttons: "提交" (Submit) and "重設" (Reset)

A calendar widget is open, showing the month of October 2016. The calendar has a header with days of the week (週日 to 週六) and a grid of dates. The date "29" is highlighted in blue.

字串轉日期時間型態資料



- 接收表單資料後接著要進行格式轉換，否則資料為字串型態，不是日期時間型態（檔案名稱：「Phpproject9」資料夾內「strtotime1.php」）。
- 表單輸入正確資料後可看見資料分屬字串與日期時間物件型態。

A screenshot of a web browser window titled "接收日期資料". The address bar shows the URL "/PhpProject9/strtotime1.php". A yellow notification bar from "NetBeans Connector" is visible. The main content area displays the output of a PHP script, showing various date and time data types and their string representations.

```
date1型態:  
string(9) "2016-12-9"  
dateTime1型態:  
object(DateTime)#1 (3) { ["date"]=> string(26) "2016-12-09  
00:00:00.000000" ["timezone_type"]=> int(3) ["timezone"]=> string(11)  
"Asia/Taipei" }  
date3型態:  
string(10) "2016-10-28"  
第1個日期為2016-12-09 12:00:00  
第2個日期為2016-12-07 11:59:59  
第一天大於第二天  
第3個日期為2016-10-28 12:00:00  
第4個日期為2016-10-25 11:59:59  
第3天大於第4天
```

字串轉日期時間型態資料



- 第12到15行將傳送過來的四個資料分別透過DateTime類別轉換產生\$dateTime1、\$dateTime2、\$dateTime3與\$dateTime4四個日期物件，第16到第19行再分別設定上述的這四個日期物件的時間。

```
12 $dateTime1 = new DateTime($date1);  
13 $dateTime2 = new DateTime($date2);  
14 $dateTime3 = new DateTime($date3);  
15 $dateTime4 = new DateTime($date4);  
16 $dateTime1->setTime(00,00,00);  
17 $dateTime2->setTime(23,59,59);  
18 $dateTime3->setTime(00,00,00);  
19 $dateTime4->setTime(23,59,59);
```

字串轉日期時間型態資料



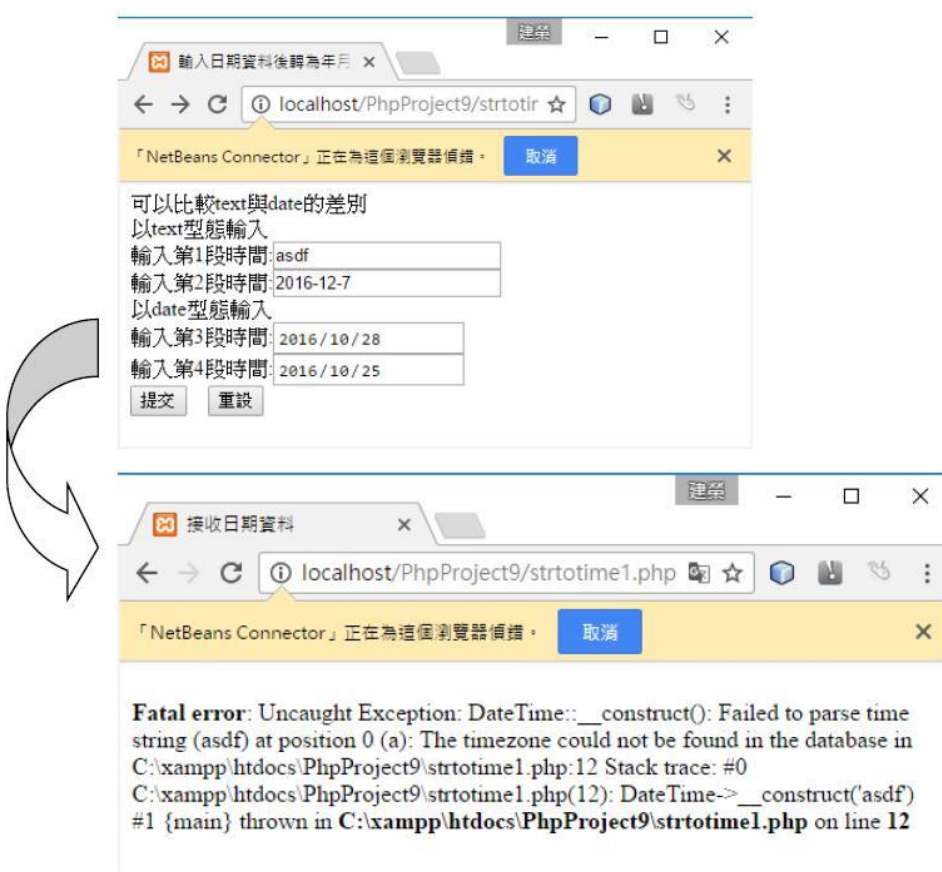
- 第21與第24行的輸出可以看到\$date1與\$date3都是字串型態的資料，雖然\$date3是date表單元件產生的資料，可是網頁傳遞過程中他是當成字串資料，而第23行可見\$dateTime11是日期時間物件型態：

```
20 echo "<br>date1 型態:<br>";  
21 var_dump($date1);  
22 echo "<br>dateTime1 型態:<br>";  
23 var_dump($dateTime1);  
24 echo "<br>date3 型態:<br>";  
25 var_dump($date3);
```

字串轉日期時間型態資料



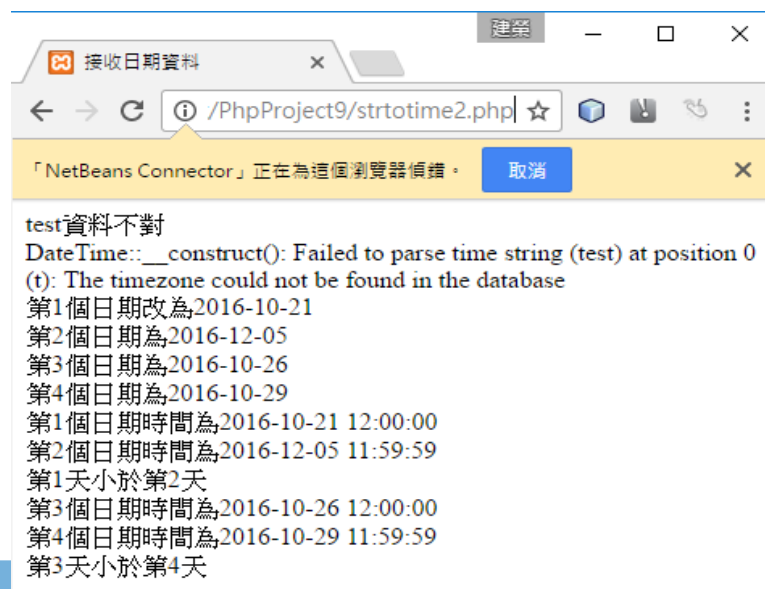
- 由於表單網頁上有兩個text型態元件輸入日期資訊，如果使用者輸入的不是日期資訊，接收資料後將會產生錯誤訊息。



字串轉日期時間型態資料



- 我們可於接收資料後將產生日期物件與資料分析語法放在try區塊內，若資料無法轉換為日期型態則拋出。
- 請將strtotime1.html另存為strtotime2.html，修改form action的內容為strtotime2.php，再請規劃建立strtotime2.php（檔案名稱：「Phpproject9」資料夾內「strtotime2.php」）。



問題與討論



- 1.請解釋甚麼是例外處理。
- 2.請設計網頁顯示2017年12月31日是禮拜幾。
- 3.請設計網頁顯示本週是一年的第幾週。
- 4.請設計網頁顯示新加坡(Asia/Singapore)與台灣的時間。
- 5.請設計網頁顯示年月日時分秒，格式可自由組合。
- 6.請設計網頁顯示明年的今日是禮拜幾。

問題與討論



- 7.以下有三題與日期時間處理有關的單選題，請您嘗試回答：
- a.與日期有關的格式化參數，請問那一個參數是錯的？
- ①大寫W代表星期幾
- ②大寫D代表星期幾
- ③小寫d代表日期
- ④大寫M代表幾月
- ⑤以上皆非

問題與討論



- b. 與時間有關的格式化參數，請問那一個參數是錯的？
- ①大寫H代表24小時制的小時
- ②小寫h代表12小時制的小時
- ③大寫l代表幾分
- ④小寫s代表幾秒
- ⑤以上皆非

問題與討論



- c. 以下PHP物件導向語法上日期時間介紹何者為非?
- ① 依據DateTime類別建立日期時間物件
- ② 物件執行format()方法進行格式化輸出
- ③ 物件執行setDate()方法設定年月日
- ④ 物件執行setTime()方法設定時分秒
- ⑤ 以上皆非

抽象類別

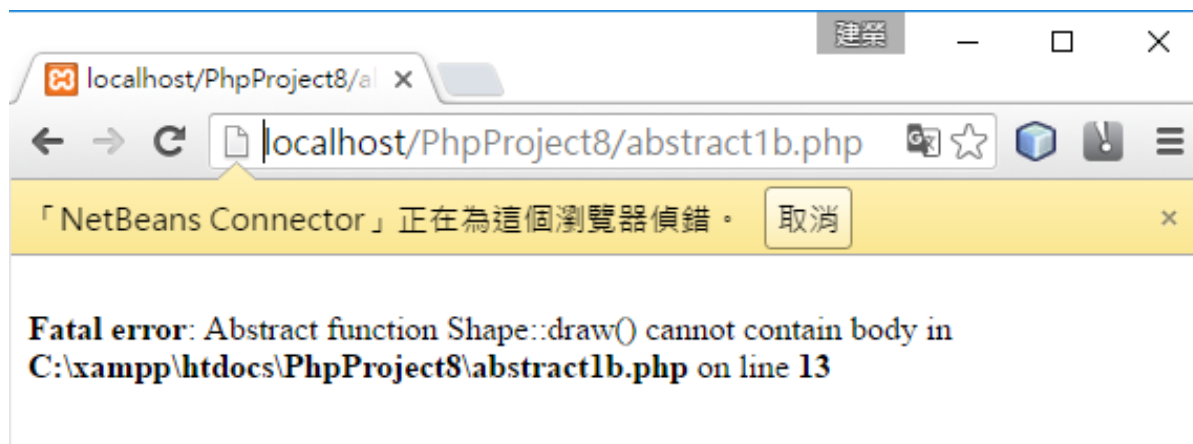


- 因為Shape類別是抽象類別，所以第35行語法由類別產生物件，就會產生錯誤訊息。
- 抽象類別不能產生物件，抽象方法內也不可以有內容。

抽象類別



- 若抽象方法內有內容，會產生什麼樣的回應呢？請由 abstract1.php 另存新檔為 abstract1b.php，在此僅列出 abstract function draw 函數語法，其他語法請參考 abstract1.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「abstract1b.php」）。



介面



- 當您規劃一個公司產品生產管理系統時，發現到C產品可以擁有A原料及B原料的多種行為方法，類別規劃時該如何進行呢？畢竟繼承只能有一個類別，而A、B、C之間也沒有繼承關係存在。
- 介面interface 是一個共同的操作環境，不同類別只要實作介面就可以操作同樣的行為，可以讓完全沒關係的數個類別來實作它，使完全不同的類別間可以有相同的部份。

介面



- 介面只定義方法，不可以宣告成員或屬性，所以介面不可以產生物件，要產生物件必須要類別去引用介面，物件才能帶入介面所定義的方法。
- 所有方法必須是public權限及abstract觀念。一個類別可以實做多個介面，一個介面也可以繼承多個介面，以擴充類別的功能。
- 介面宣告的語法為「interface 介面名稱」，而類別實做介面的語法為「class 類別名稱 implements 介面名稱」。

介面

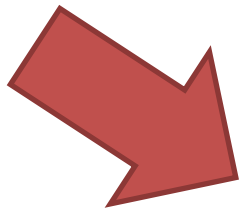


- 飛機與汽車並無親屬關係，但兩者都是機械的一種，所以類別內可設計一個機械的介面，提供setname與operation兩個方法給類別引用後實作。
- 如何於PHP內表達飛機與汽車的資訊呢？

```
07 interface machine
08 {
09     public function setname($name);
10     public function operation( );
11 }
```



```
12 class plain implements machine
13 {
22 }
```

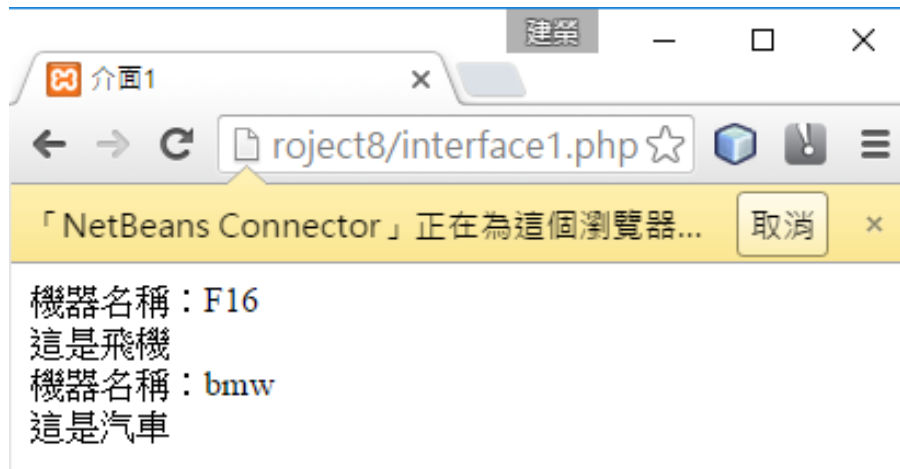


```
23 class car implements machine
24 {
33 }
```

介面



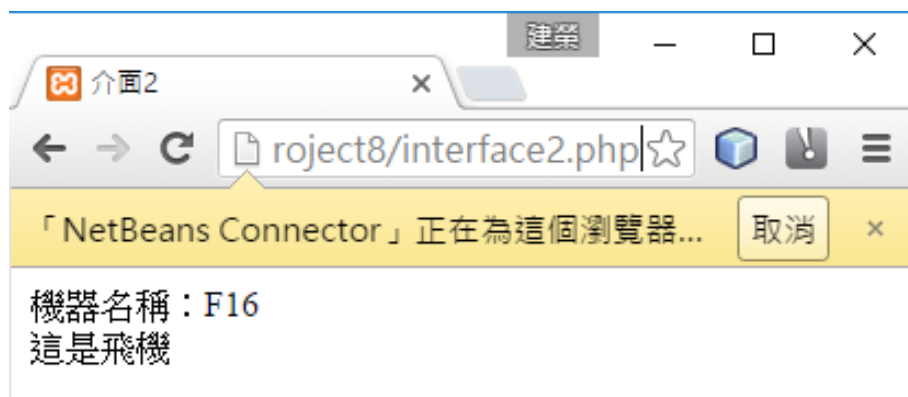
- 請練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface1.php」）。



介面



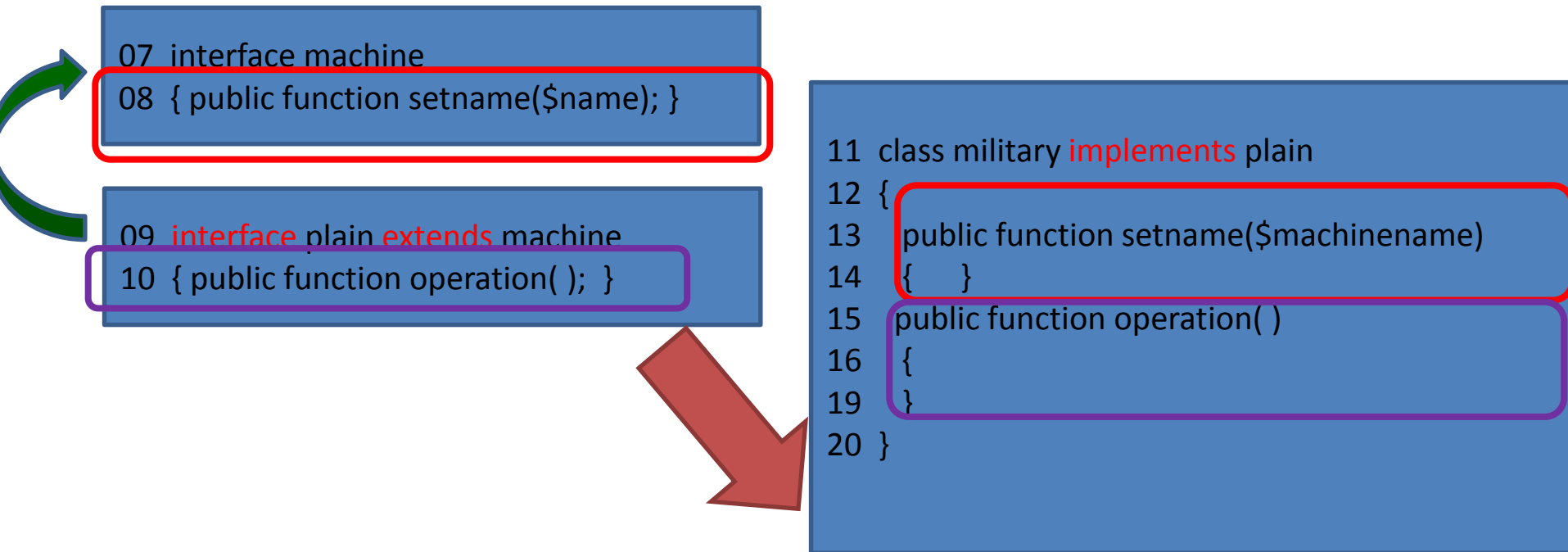
- 介面之間也可使用繼承的方式擴充功能，而類別再透過引用子介面方式擁有這兩個介面提供的功能。
- 請依練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface2.php」）。



介面



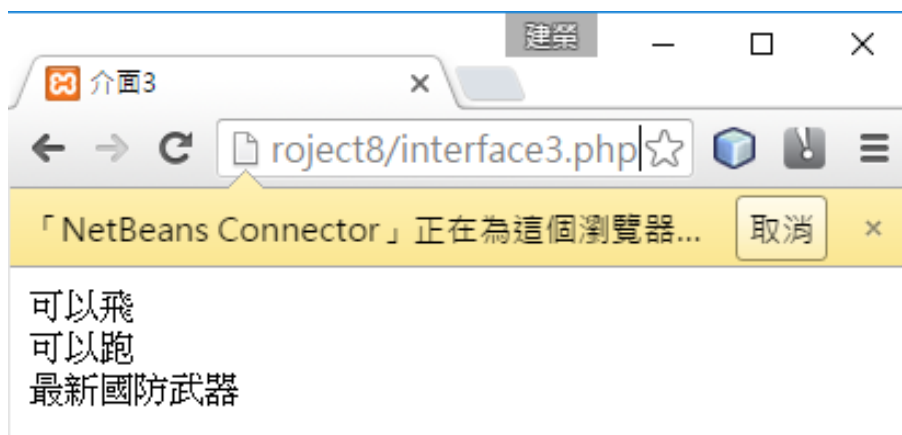
- 請依練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface2.php」）。



介面



- 介面可以繼承多個介面，例如army介面想要繼承plain與car兩個介面，請依以下練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface3.php」）。



介面



- 介面可以繼承多個介面，例如army介面想要繼承plain與car兩個介面，請依以下練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「interface3.php」）。

```
07 interface plain
08 { public function fly( );}
```

```
09 interface car
10 { public function run( ); }
```

```
11 interface army extends plain , car
12 { public function fire( ); }
```

```
13 class arms implements army
14 {
15     public function fly( )
16     { echo "可以飛"."<br>"; }
17     public function run( )
18     { echo "可以跑"."<br>"; }
19     public function fire( )
20     { echo "最新國防武器"."<br>"; }
21 }
```



多型特性



- 多型指呼叫名稱相同的函數，卻在不同情況下有不同的表現。
- 多型包括了方法過載 (Overloading) 與方法蓋寫 (Overriding) 兩種，不過PHP不支援方法過載。

方法過載



- 方法過載指類別內可定義多個同名的方法，依方法接收參數的類型與數量不同來判斷如何執行。
- 但PHP並不支援方法過載，所以PHP類別內請勿設計兩個同名的方法。

方法過載



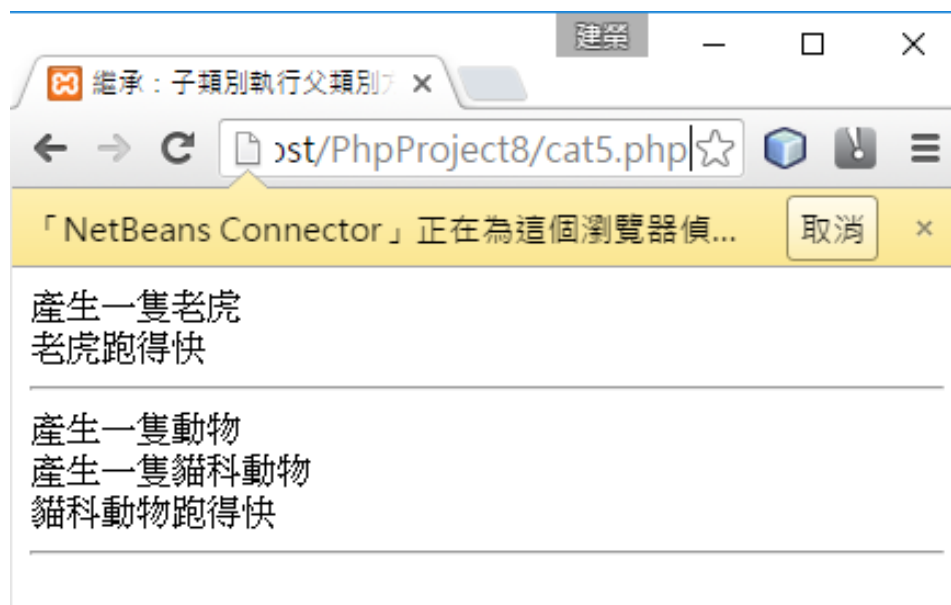
- 若類別內有兩個同名方法，網頁會如何呈現呢？請由 box4.php 另存新檔為 box6.php，在此僅列出兩個同名的方法語法，其他語法請參考 box4.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「box6.php」）。

```
11 function __construct($a,$b,$c) {  
12     $this->length1=$a;  
13  
14     Method or function "getdata" has already been declared  
15     ----  
16     (Alt-Enter shows hints)  
17 function getdata() {  
18     return $this->breadth1*$this->height1*$this->length1;  
19 }  
20 function getdata($a,$b) {  
21     return $this->breadth1*$this->height1*$this->length1;  
22 }
```

方法覆寫



- 當您不滿意由父類別那裡繼承到的方法時，您可以寫一個同名方法，就可將他覆蓋。
- 請依練習觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat5.php」）。



方法覆寫



- cat類別是tiger類別的父類別，而cat類別於第13行設計了run()方法，tiger類別於第19行也設計了run()方法。
- 當第27行由tiger類別產生的物件執行run()方法，您會發現到系統會執行tiger類別設計的run()方法，而不是父類別所提供的方法。

方法覆寫



- 所以當父子類別都有相同的方法時，依子類別產生的物件會執行子類別的方法，不會執行父類別的方法。

```
08 class cat extends animal {  
09   public function __construct( )  
13   public function run( )  
15 }
```

```
16 class tiger extends cat{  
17   public function __construct( )  
19   public function run( )  
21 }
```



```
26 $tiger1 = new tiger;  
27 $tiger1->run( );
```

方法覆寫



- 父類別內方法若為public權限，子類別產生的物件想要執行父類別的方法，有兩種方式。
- 第一種是子類別內不要有同名方法，子類別產生的物件就會直接執行父類別內的方法，請由cat5.php另存新檔為cat6.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat6.php」）。

方法覆寫

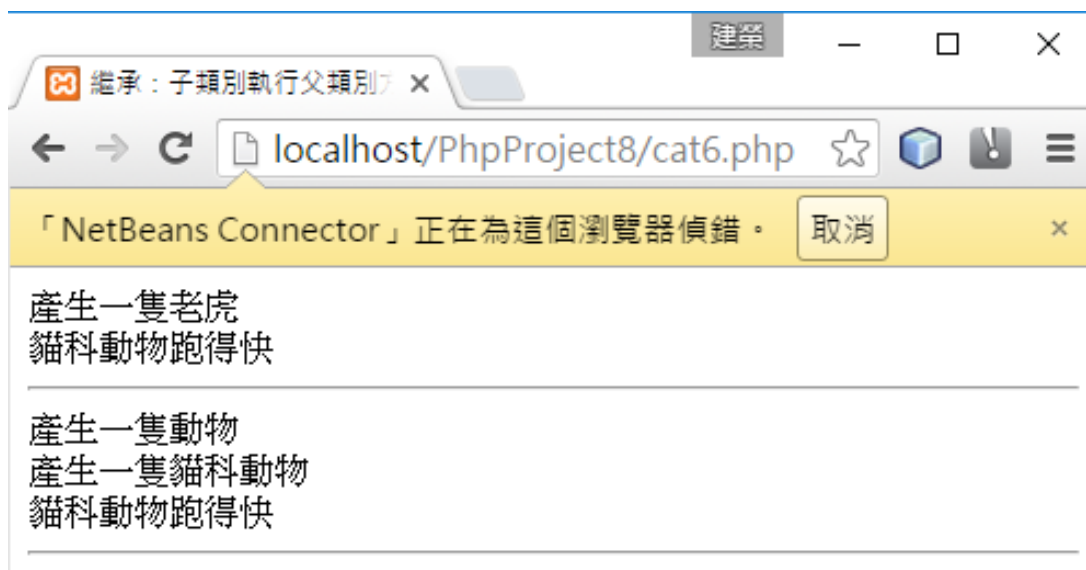


- 父類別內方法若為public權限，子類別產生的物件想要執行父類別的方法，有兩種方式：
- 第一種是子類別內不要有同名方法，子類別產生的物件就會直接執行父類別內的方法。

方法覆寫



- 請由cat5.php另存新檔為cat6.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat6.php」）。



方法覆寫



- 請由cat5.php另存新檔為cat6.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat6.php」）。

```
08 class cat extends animal {  
09     public function __construct( )  
13     public function run( )  
15 }
```

```
16 class tiger extends cat{  
17     public function __construct( )  
  
21 }
```

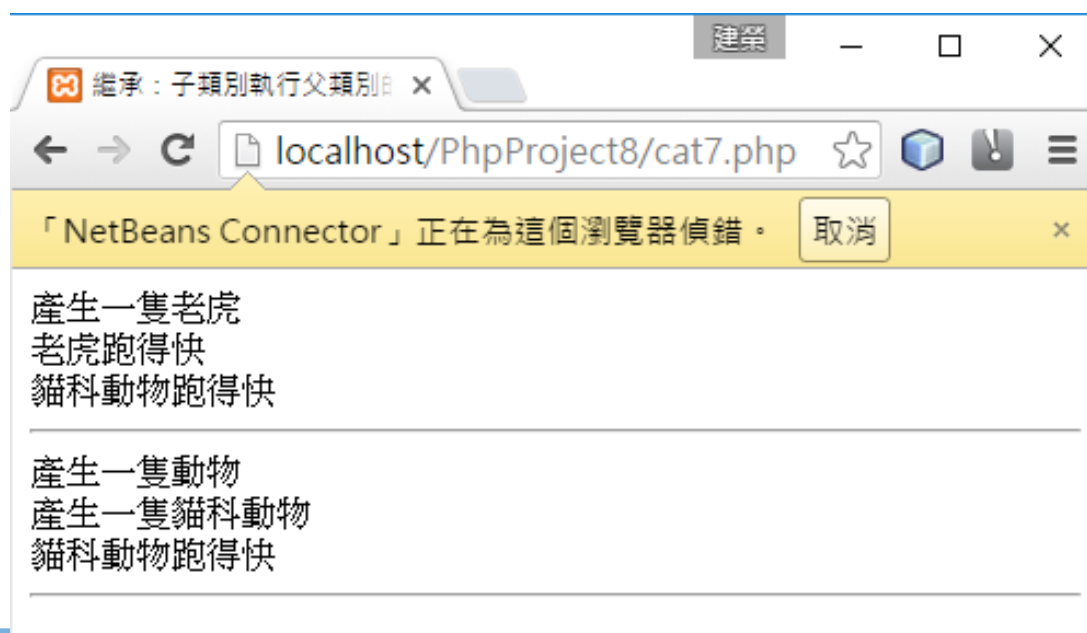


```
26 $tiger1 = new tiger;  
27 $tiger1->run( );
```


方法覆寫



- 第二種是子類別內以「parent::方法名稱()」方式呼叫，子類別產生的物件就會執行父類別內的方法。
- 請由cat5.php另存新檔為cat7.php，在此僅列出tiger類別內的語法，其他語法請參考cat5.php，再請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「cat7.php」）。



方法覆寫



- cat類別是tiger類別的父類別，而cat類別於第13行設計了run()方法，tiger類別於第19行也設計了run()方法。
- tiger類別於run()方法內執行「parent::run()」語法(第22行)，執行父類別的run()方法。

方法覆寫



- 當tiger類別產生的物件執行run()方法，您會發現到系統除執行子類別，也就是tiger類別的run()方法外，也執行父類別所提供的方法。

```
08 class cat extends animal {  
09     public function __construct( )  
13     public function run( )  
15 }
```

```
16 class tiger extends cat{  
17     public function __construct( )  
19     public function run( )  
20     {  
21         echo "老虎跑得快". "<br>";  
22         return parent::run( );  
23     }
```

```
26 $tiger1 = new tiger;  
27 $tiger1->run( );
```

新增物件導向功能

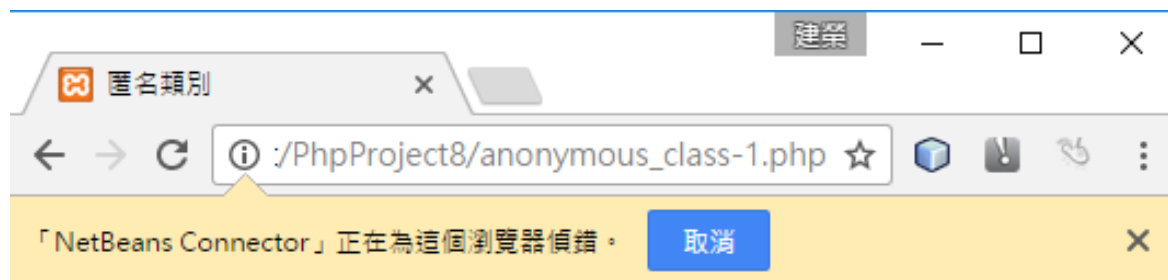


- PHP7於物件導向這部分功能如下：
 - 新增匿名類別
 - 修改use方法
 - 新增call方法
 - 新增例外處理
- 新增的例外處理於10-1節介紹，而call方法將於第11-3-3節介紹，本章將介紹其他項目。

新增匿名類別



- PHP7支援匿名類別，這種類別沒有類別名稱，直接使用new方式建立物件。
- 通常用於只需要使用一次物件的情況下進行。請觀察網頁執行結果（檔案名稱：「PhpProject8」資料夾內「anonymous_class-1.php」）。



修改use方法



- PHP可以使用載入其他網頁的namespace命名空間內的常數、函數、類別與靜態方法，在此我們規劃了第一個library檔案（檔案名稱：「PhpProject8」資料夾內「lib1.php」）。
- 接著規劃第二個library檔案（檔案名稱：「PhpProject8」資料夾內「lib2.php」）。

修改use方法



- 於PHP7之前的版本您得使用多次的use語法才能呼叫後使用（檔案名稱：「PhpProject8」資料夾內「uselib.php」）。

```
05 use App\Lib1\MyClass as C1;  
06 use App\Lib1\MyClass2 as C2;  
07 use App\Lib2\MyClass as C3;  
08 use function App\Lib1\MyFunction as Cf1;  
09 use function App\Lib1\MyFunction2 as Cf2;  
10 use function App\Lib2\MyFunction as Cf3;  
11 use const App\Lib1\MYCONST1 as Cont1;  
12 use const App\Lib1\MYCONST2 as Cont2;  
13 use const App\Lib2\MYCONST as Cont3;
```

修改use方法



- PHP7可以使用一個use語法導入同一個命名空間內的類別、方法與常數（檔案名稱：「PhpProject8」資料夾內「uselib2.php」）。

```
05 use App\Lib1\{MyClass as C4,MyClass2 as C5};  
06 use function App\Lib1\{MyFunction as Cf4,MyFunction2 as Cf5};  
07 use const App\Lib1\{MYCONST1 as Cont4,MYCONST2 as Cont5};
```


問題與討論



- 1.如何呼叫父類別的建構子與解構子?
- 2.如何使用父類別內的成員?
- 3.請問抽象類別可以產生物件嗎?
- 4.請問抽象方法可以有內容嗎?
- 5.請問子類別一定要實作抽象方法嗎?
- 6.何謂介面?
- 7. PHP支援方法過載嗎?
- 8.請解釋甚麼是匿名類別。