

```

include("langs.php");

//顯示 tr_strings_en 或 tr_strings_tw 的 string_id 為 FIRST_STRING
//的文字
echo "<hr>" . $tr->gstr("FIRST_STRING", array("Wojtek")) .
"<hr>";

//顯示 tr_strings_en 或 tr_strings_tw 的 string_id 為 SECOND_STRING
//的文字
echo $tr->gstr("SECOND_STRING");

//可連結到第二頁的超連結
echo "<br><a href=second.php?Lang=" . (isset($_GET['Lang']) ?
$_GET['Lang'] : 'pl') . ">" . $tr->gstr("LINK",
array($tr->gstr("SECOND"))) . "</a>";
echo "</body></html>";
?>

```

被引入的網頁 langs.php 內容如下，提供語言的切換功能：

```

<?php
//抓取在 first.php 中已設定的語言在 tr_langsavail 資料表的資料
$langsarray = $tr->getOtherLangs();
if (DB::isError($langsarray))
die ($langsarray->getMessage());

//建立一個語言切換的裝置
while(current($langsarray)) {
echo " <a href=\" . $_SERVER["PHP_SELF"] . "?Lang=" .
current(current($langsarray));
    foreach($_GET as $key=>$value)
        if ($key != "Lang") echo "&$key=$value";

```

```
echo "\">" . next(current($langsarray)) . "</a>";
next($langsarray);
}
?>
```

上述範例網頁執行的結果：



圖三十九 Translate 套件範例網頁以英文顯現

預設為英語網頁，若按「繁體中文」會切換到中文網頁。



圖四十 Translate 套件範例網頁以繁體中文顯示

請注意，first.php 後的?Lang=tw 讓 PHP 作為識別使用者所使用的語言形式，若沒有?Lang=xx 則會使用預設的語言顯示網頁。

5-13 PHPUnit 套件

PHPUnit 套件是一個用在類別與函式除錯的工具。一般人撰寫程式碼，除錯的方式都是在程式中加入 `echo()`，`var_dump()` 等函式，但這種方式相當費時費力。用 PHPUnit 針對以下情況而言是一種最好的方法：

- 設計類別或應用程式
- 建立測試環境
- 除錯

底下利用官方網站提供的手冊中 Martin Jansen，Alexander Merz(2004)兩位所撰寫的範例程式來加以說明 PHPUnit 套件的使用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

第一個檔案名為 `string.php` 內含類別 `String`

```
<?php
class String
{
    //contains the internal data
    var $data;

    //建構子
    function String($data) {
        $this->data = $data;
    }

    // 建立一個文字物件的複製
    function copy() {
        $ret = new String($this->data);
        return $ret;
    }

    //增加其他的文字物件到類別中的方法
    function add($string) {
```

```

        $this->data = $this->data.$string->toString("%ss");
    }

    //傳回一個格式化的字串
    function toString($format) {
        $ret = sprintf($format, $this->data);
        return $ret;
    }
}
?>

```

接著建立一個測試類別，此類別可以檢查 String 類別中的每一個函式，這個網頁含有一個類別名為 StringTest 是繼承自 PHPUnit_TestCase 類別。它包含測試的函式，這些函式都以「test」為開頭文字，他們會去比較預期值與結果值，然後經由 assert*() 函式來決定是通過檢測還是檢測失敗。

```

<?php
//---- 此檔案命名為 testcase.php ----
<?php
require_once 'string.php';
//引入 PJUnit.php 檔

require_once 'PHPUnit.php';
//建立 StringTest 類別，繼承自 PHPUnit_TestCase 類別

class StringTest extends PHPUnit_TestCase
{
    var $abc;
    // 建構子
    function StringTest($name) {
        $this->PHPUnit_TestCase($name);
    }
}

```

```

}

//在執行檢測前所呼叫的函式
// 此函式在 PHPUnit_TestCase 有定義，在此改寫
function setUp() {
    // 用字串 abc 建立一個 String 物件。
    $this->abc = new String("abc");
}

// 在執行測試後所呼叫的函式
//此函式在 PHPUnit_TestCase 有定義，在此改寫
function tearDown() {
    // 刪除物件
    unset($this->abc);
}

// 測試 String 類別中的 ToString 函式
function testToString() {
    $result = $this->abc->toString('contains %s');
    //希望得到的執行結果
    $expected = 'contains abc';
    $this->assertTrue($result == $expected);
}

//測試 String 類別中的 Copy 函式
function testCopy() {
    $abc2 = $this->abc->copy();
    $this->assertEquals($abc2, $this->abc);
}

//測試 String 類別中的 Add 函式

```

```
function testAdd() {
    $abc2 = new String('123');
    $this->abc->add($abc2);
    $result = $this->abc->toString("%s");
    $expected = "abc123";
    $this->assertTrue($result == $expected);
}
}
?>
```

最後建立一個執行測試的網頁，名為 stringtest.php，執行此網頁即可除錯

```
<?php

require_once 'testcase.php';
require_once 'PHPUnit.php';

$suite = new PHPUnit_TestSuite("StringTest");
$result = PHPUnit::run($suite);

echo $result -> toString();
?>
```

執行結果如下：

```
TestCase stringtest->testtostring() passed
TestCase stringtest->testcopy() passed
TestCase stringtest->testadd() failed: expected true, actual
false
```

此結果代表 add() 函式有錯誤，經過檢查原來是 %s 寫成了 %ss，將它修改後，再執行一次測試，結果如下

```
TestCase stringtest->testtostring() passed
TestCase stringtest->testcopy() passed
TestCase stringtest->testadd() passed
```

大功告成，String 類別的函式都正確了。

5-14 HTML Treemenu 套件

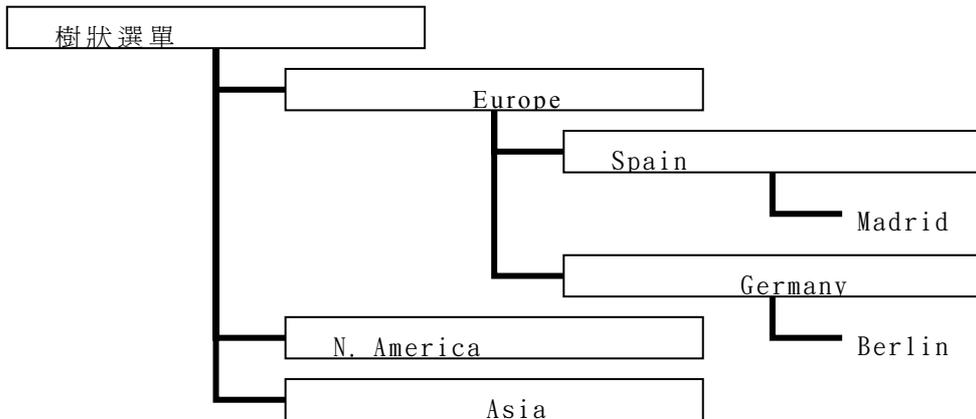
許多人想在網頁中做動態的樹狀選單，可以用此套件讓這個目的很容易就達到。Demian Turner(2002)認為只要以下二個步驟即可，1. 建立選單項目的資料庫. 2. 撰寫程式碼。

5-14.1 建立選單項目的資料庫

利用以下的 mysql 資料表 cattest 的例子來說明：(Demian Turner，2002)

id	name	parent	link
1	Europe	0	test.php
2	N. America	0	test1.php
3	Asia	0	test2.php
4	Spain	1	test3.php
5	Madrid	4	test4.php
6	Germany	1	test5.php
7	Berlin	6	test6.php

將所有選單的項目，包括主項目與次項目，用 id 代表項目編號，而次項目用 parent 來代表其對應的主項目的 id 的編號，而 link 代表超連結的網頁所在。上例中，主項目與次項目的對應應如下：



5-14.2 撰寫程式碼

```

<?php
//引入 PHPguru' s Tree 類別的網頁，該類別尚未放入 PEAR 中
//須從 http://www.phpguru.org/tree.html 下載來安裝
include 'Tree.php';

//使用 PHPguru' s Tree 類別中的 createFromMySQL 方法來從資料庫
//的擷取項目的資料
$tree = &Tree::createFromMySQL(array('host'      => 'localhost' ,
                                     'user'       => 'root' ,
                                     'pass'       => '' ,
                                     'database'   => 'testdata' ,
                                     'query'      => 'SELECT
id , parent AS parent_id , name AS text , link FROM cattest ORDER BY
parent_id , id' ));
?>
<?php
//引入 TreeMenu 套件的 TreeMenu.php 檔
include 'TreeMenu.php';
  
```

```

//設定樹狀選單的選項，如：外顯文字，資料夾的圖片…等。
//放入一個名為$nodeOptions的陣列中
$nodeOptions = array(
    'text'           => '' ,
    'link'           => '' ,
    'icon'           => 'folder.gif' ,
    'expandedIcon'  => 'openfoldericon.png' ,
    'class'          => '' ,
    'expanded'      => false ,
    'linkTarget'    => '_self' ,
    'isDynamic'     => 'true' ,
    'ensureVisible' => '' ,
);

//再將一些樹狀選單的設定項目放入$options陣列中
$options = array(
    'structure' => $tree ,
    'type'      => 'heyes' ,
    'nodeOptions' => $nodeOptions);
?>
<?php

//用 createFromStructure 方法建立一個 HTML_TreeMenu 物件
$menu = &HTML_TreeMenu::createFromStructure($options);
?>
<?php

//最後將 HTML_TreeMenu 物件作成一個 DHTML 的可摺疊的選單
$treeMenu = &new HTML_TreeMenu_DHTML($menu ,
    array( 'images' => 'imagesAlt2' ,
          'defaultClass' => 'treeMenuDefault' ));
?>

```

```

<html>
<head>
  <style type="text/css">
    .treeMenuDefault {
      font-style: italic;
    }
  </style>
  <script src="TreeMenu.js" language="JavaScript"
type="text/javascript"></script>
</head>
<body>
<?$treeMenu->printMenu()?>
</body>
</html>

```

產生的範例結果如下圖：



圖四十一 樹狀選單形式圖(一)

另外一種不用在資料庫先建立項目資料表，完全透過 javascript 碼，也可產生動態樹狀選單的方法，範例程式碼如下

```

<html><head>

//設定 CSS 樣式

```

```

<style type="text/css">
  body {
    font-family: Georgia;
    font-size: 11pt;
  }
  .treeMenuDefault {
    font-style: italic;
  }
  .treeMenuBold {
    font-style: italic;
    font-weight: bold;
  }
</style>

```

```
//引入 TreeMenu. js 檔
```

```

  <script src="TreeMenu.js" language="JavaScript"
type="text/javascript"></script>

```

```
</head>
```

```
<body>
```

```
<script language="javascript" type="text/javascript">
```

```

//建立一個 HTML_TreeMenu 物件，參數依序為「圖檔所在路徑」，「物件
//名稱」，「超連結的目標」，「預設的 CSS 的 class」，「在
//client 端執行」，「無上層圖檔」

```

```

objTreeMenu_1 = new TreeMenu("../guruTree/imagesAlt2",
"objTreeMenu_1", "_self", "treeMenuDefault", true, false);

```

```

//建立第一個節點，即最上層的節點，並放入 TreeMenu 物件中成為一個
//項目，參數依序為「外顯文字」，「圖檔名稱」，「超連結檔案」，
//「是否為展開的」，「是否為動態」，「CSS 類別」，「超連結的目
//標」，「擴展後的圖檔」。

```

```

newNode = objTreeMenu_1.addItem(new TreeNode('First level' ,
'folder.gif' , 'test.php' , false , true , '' , '' ,
'folder-expanded.gif' ));

//設定 newNode 點選 First level 字樣後會觸發的事件
newNode.setEvent('onclick' , 'alert(\`foo\`); return false');

//設定 newNode 點選 onexpand 圖案後會觸發的事件
newNode.setEvent('onexpand' , 'alert(\`Expanded\`)' );

//建立第二個節點，並放入第一個節點中成為一個項目
newNode_1 = newNode.addItem(new TreeNode('Second level' ,
'folder.gif' , 'test.php' , false , true , '' , '' ,
'folder-expanded.gif' ));

//建立第三個節點，並放入第二個節點中成為一個項目
newNode_1_1 = newNode_1.addItem(new TreeNode('Third level' ,
'folder.gif' , 'test.php' , false , true , '' , '' ,
'folder-expanded.gif' ));

//建立第四個節點，並放入第三個節點中成為一個項目
newNode_1_1_1 = newNode_1_1.addItem(new TreeNode('Fourth level' ,
'folder.gif' , 'test.php' , false , true , '' , '' ,
'folder-expanded.gif' ));

//建立第五個節點，並放入第四個節點中成為一個項目
newNode_1_1_1_1 = newNode_1_1_1.addItem(new TreeNode('Fifth
level' , 'folder.gif' , 'test.php' , false , true ,
'treeMenuBold' , '' , 'folder-expanded.gif' ));
newNode_2 = newNode.addItem(new TreeNode('Second level , item 2' ,
'folder.gif' , 'test.php' , false , true , '' , '' ,

```

```

' folder-expanded.gif' ));
newNode_3 = newNode.addItem(new TreeNode(' Second level , item 3' ,
' folder.gif' , ' test.php' , false , true , '' , '' ,
' folder-expanded.gif' ));
newNode = objTreeMenu_1.addItem(new TreeNode(' First level' ,
' folder.gif' , ' test.php' , false , true , '' , '' ,
' folder-expanded.gif' ));
newNode.setEvent(' onclick' , ' alert(\` foo\`); return false' );
newNode.setEvent(' onexpand' , ' alert(\` Expanded\`)' );
newNode_1 = newNode.addItem(new TreeNode(' Second level' ,
' folder.gif' , ' test.php' , false , true , '' , '' ,
' folder-expanded.gif' ));
newNode_1_1 = newNode_1.addItem(new TreeNode(' Third level' ,
' folder.gif' , ' test.php' , false , true , '' , '' ,
' folder-expanded.gif' ));
newNode_1_1_1 = newNode_1_1.addItem(new TreeNode(' Fourth level' ,
' folder.gif' , ' test.php' , false , true , '' , '' ,
' folder-expanded.gif' ));
newNode_1_1_1_1 = newNode_1_1_1.addItem(new TreeNode(' Fifth
level' , ' folder.gif' , ' test.php' , false , true ,
' treeMenuBold' , '' , ' folder-expanded.gif' ));
newNode_2 = newNode.addItem(new TreeNode(' Second level , item 2' ,
' folder.gif' , ' test.php' , false , true , '' , '' ,
' folder-expanded.gif' ));
newNode_3 = newNode.addItem(new TreeNode(' Second level , item 3' ,
' folder.gif' , ' test.php' , false , true , '' , '' ,
' folder-expanded.gif' ));

//將整個樹狀結構繪出。
objTreeMenu_1.drawMenu();

```

```
</script>
</body>
</html>
```

執行結果如下圖：



圖四十二 樹狀選單形式圖(二)

5-15 HTTP_Upload 套件

這個類別提供一個用 HTML 表單就可以上傳檔案的系統，它具有以下幾個特色：

- 可處理一個或多個檔案
- 可從暫存檔安全的拷貝到目的檔-檔案的上傳是先傳送成暫存檔，待傳送完成再拷貝為目的檔，並將暫存檔刪除。
- 可簡單的偵測是否成功的上傳或檔案遺失或發生錯誤
- 支援許多種語言的錯誤訊息

底下利用 Tomas V.V.Cox(2001)所撰寫的範例程式來加以說明 HTTP_Upload 套件的用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

第一個範例:上傳條件為，一次只能上傳一個檔案，可傳空檔案

```

index.htm
-----
<?php
<form action="./files.php" enctype="multipart/form-data">
```

```
File 1: <input type="file" name="userfile"><br>
<input type="submit" name="submit" value="Upload!">
</form>
?>
```

處理上傳檔的程式碼檔案 files.php 如下：

```
files.php
-----
<?php
//引入 Upload.php 檔
require 'HTTP/Upload.php';
$upload = new HTTP_Upload(); //建立一個 HTTP_Upload 物件
$file = $upload->getFiles('userfile'); //傳回檔案物件或錯誤物件
件
if (PEAR::isError($file)) { //若傳回的是錯誤物件
    die ($file->getMessage());
}

//檢查上傳是否成功
if ($file->isValid()) {
//這個方法若成功上傳到./uploads_dir/下，將會傳回你上傳的檔名
    $file_name = $file->moveTo('./uploads_dir/');
    if (PEAR::isError($file_name)) {
        die ($file_name->getMessage());
    }
}
?>
```

第二個範例：上傳條件為，一次只能上傳一個檔案，不可傳空檔。

```
<?php
require 'HTTP/Upload.php';
```

```

//設定當錯誤發生時當處理方式為停止再執行程式。

PEAR::setErrorHandler (PEAR_ERROR_DIE);
$upload = new HTTP_Upload();
$file = $upload->getFiles('userfile');

//檢查是否成功上傳
if ($file->isValid()) {
    $file_name = $file->moveTo('./uploads_dir');
// 若使用者傳送空檔案則用以下方式偵測
} elseif ($file->isMissing()) {
    die ('You must upload one file');
// 若使用者非傳送空檔案，但上傳時發生錯誤
} else {
    die ('There were errors uploading ->' . $file->errorMsg());
}
?>

```

第三個範例：上傳條件為，一次只能上傳一個檔案，並將上傳檔改檔名。

```

<?php
require 'HTTP/Upload.php';
$upload = new HTTP_Upload();
$file = $upload->getFiles('userfile');
if ($file->isValid()) {
    /*
    * $mode 可以設定成以下參數：
    * - 'uniq' ->將檔名改成獨一無二的名稱，如 42341242353451.jpg
    * - 'safe' ->若檔名不正確會改為正確，如'p 塿!.jpg'會改
    *           為'p_a__.jpg'
    * - 'real' -> 保留原始檔名
    */
}

```

```

* - 其他值 -> 就以這個值當做檔名，如 abc.jpg
*
* $prepend 在檔名前可加上此字串
* $append 在檔名後可加上此字串
*/
$file->setName ($mode, $prepend, $append);
$file_name = $file->moveTo('./uploads_dir');
}
?>
<?php
require 'HTTP/Upload.php';

//設定當錯誤發生時當處理方式為停止再執行程式。
PEAR::setErrorHandler (PEAR_ERROR_DIE);
$upload = new HTTP_Upload();
$file = $upload->getFiles('userfile');

//檢查是否成功上傳
if ($file->isValid()) {
    $file_name = $file->moveTo('./uploads_dir');
// 若使用者傳送空檔案則用以下方式偵測
} elseif ($file->isMissing()) {
    die ('You must upload one file');
// 若使用者非傳送空檔案，但上傳時發生錯誤
} else {
    die ('There were errors uploading ->' . $file->errorMsg());
}
?>

```

第四個範例：上傳條件為，一次能上傳多個檔案。

```
<form action="files.php" enctype="multipart/form-data">
```

```
Image 1: <input type="file" name="userfile[]"><br>
Image 2: <input type="file" name="userfile[]"><br>
Image 3: <input type="file" name="userfile[]"><br>
<input type="submit" name="sub" value="Upload!">
</form>
```

處理上傳檔案的 files.php

```
<?php
...
$files = $upload->getFiles(); // 傳回檔案物件的陣列或錯誤物件

//利用回圈來將檔案一一上傳
foreach ($files as $file) {
    if ($file->isValid()) {
        ...
    }
}
?>
```

也可以用其他的陣列形式來上傳檔案，例如：

```
//檔名用不同的陣列形式
Image 1: <input type="file" name="userfile[aaa]"><br>
Image 2: <input type="file" name="userfile[]"><br>
Image 3: <input type="file" name="userfile[]"><br>
Image 4: <input type="file" name="otherfile"><br>
```

處理檔案上傳的機制部分可改寫為：

```
<?php
$upload = new HTTP_Upload;
$file1 = $upload->getFiles('userfile[aaa]');
$file2 = $upload->getFiles('userfile[0]');
```

```
$file2 = $upload->getFiles('userfile[1]');
$file3 = $upload->getFiles('otherfile');
?>
```

若想知道上傳檔案的總總屬性可用 `getProp` 方法來得知

```
<?php
...
$upload = new HTTP_Upload;
$file = $upload->getFiles('userfile');

//得到所有的上傳檔案的屬性並放入一個$properties 陣列
$properties = $file->getProp();

/*屬性陣列範例如下：
array(
    'real'      => 'myphoto.jpg' ,      //真實上傳檔案檔名
    'name'      => '12312334234.jpg' , //上傳後的檔名
    'form_name' => 'userfile' ,        //input 元件的名稱
    'ext'       => 'jpg' ,              //副檔名
    'tmp_name'  => '/tmp/Ace234d_da4' , //在 tmp 目錄下的暫存檔
    'size'      => '230' ,              //檔案大小
    'type'      => 'image/jpg' ,        //檔案的 mime 型態
    'error'     => null                  // 錯誤， for ex:
                                     // NO_USER_FILE -> 代表未上傳檔案
                                     // TOO_LARGE   -> 代表檔案大小超過上限
);
*/

//若只想得知某樣屬性，例如想知道上傳檔的副檔名，範例如下：
$extension = $file->getProp('ext');
?>
```

5-16 Mail 套件

一般使用者最常用到的網站應用程式可能是電子郵件系統，而 PEAR 讓 mail 變得更簡單，它提供了一個郵件傳送抽象層(mail transport abstraction)，可以送信不用經過複雜的透過指令存取 smtp 的過程，而且提供比函式更多的功能。(30)

這個套件如何使用呢？套件內的類別有名為 mail 的父類別(parent class)，和其他子類別(child class)，分別有 Mail_mail，Mail_smtp，Mail_sendmail 等。父類別大多提供一些工廠方法(factory method)，可用來建立子類別的物件，這些子類別的物件都可以用 send()方法來寄信並回應原呼叫的程式。雖然每個子類別需要的參數會有少許不同。

電子郵件中最困難的就是處理附加檔案，可能最簡單的處理方式是讓另一個類別來處理它。幸好，PEAR 對處理這個問題的能力很強大。有兩個類別可供使用，一個可用來建立含附件的電子郵件，另一個類別可用來解碼。名為 Mail_mime 的類別當建立完後可用 setTXTBody()，setHTMLBody()，addHTMLImage()，addAttachment()這些方法來建立新郵件，然後用 get()和 headers()方法來建立郵件並傳回結果，額外有一個優點是這兩個方法所傳回的資料可以直接傳到抽象層的 send()方法。另外，不同的參數可用其他的方法來提供，例如編碼方式或字元型態。如果你想用快速又簡單的方法來傳送複雜的郵件，這個 Mail_mime 類別是非常有用的。

但如果需要更進階的功能，就需要 Mail_mimepart 類別了。這個類別的每個物件代表一個附件，物件需提供詳細的 mime 訊息給建構子，包括：內容型態，編碼，實際的內容等。另外還有兩個方法為 encode()與 addsubpart()可用來建立更複雜的結構。

底下利用 Richard Heyes (2002)所撰寫的範例程式來加以說明 Mail 套件的用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

第一個範例檔，運用 PHP 的 mail()方法來寄信

```
<?php
//引入 Mail.php 檔
```

```
include('Mail.php');

//設定收信人
$recipients = 'joe@example.com';

//設定送信人，收信人，標題
$headers['From'] = 'richard@phpguru.org';
$headers['To'] = 'joe@example.com';
$headers['Subject'] = 'Test message';

//設定信件內容
$body = 'Test message';

//用 Mail::factory 方法建立 mail 物件
$mail_object =& Mail::factory('mail');

//用 mail 物件的 send()方法來寄信
$mail_object->send($recipients, $headers, $body);
?>
```

第二個範例檔，運用 sendmail 的執行檔來寄信

```
<?php
include('Mail.php');
$recipients = 'joe@example.com';
$headers['From'] = 'richard@phpguru.org';
$headers['To'] = 'joe@example.com';
$headers['Subject'] = 'Test message';
$body = 'Test message';

//設定伺服器 sendmail 的執行檔所在路徑
$params['sendmail_path'] = '/usr/lib/sendmail';
```

```
//運用 Mail::factory 方法來建立 sendmail 物件，並以$params 為參數
$mail_object =& Mail::factory('sendmail', $params);

//用 sendmail 物件的 send()方法來寄信
$mail_object->send($recipients, $headers, $body);
?>
```

第三個範例檔，運用 SMTP 來寄信

```
<?php
include('Mail.php');
$recipients = 'joe@example.com';
$headers['From'] = 'richard@phpguru.org';
$headers['To'] = 'joe@example.com';
$headers['Subject'] = 'Test message';
$body = 'Test message';

//設定郵件主機名稱
$params['host'] = 'mail.example.com';

//用 Mail::factory 方法來建立 mail 物件
$mail_object =& Mail::factory('smtp', $params);

//用 mail 物件的 send()方法來寄信
$mail_object->send($recipients, $headers, $body);
?>
```

第四個範例檔，在信中附加檔案

```
<?php
include('Mail.php');
```

```
//引入 Mail 套件中的 mime.php 檔
include('Mail/mime.php');

//信若以文字模式的內容
$text = 'Text version of email';

//信若以 HTML 模式的內容
$html = '<html><body>HTML version of email</body></html>';

//附加檔案的路徑與檔名
$file = '/home/richard/example.php';

//換行符號
$CrLf = "\r\n";

//寄件者與標題所組成的陣列
$headers = array(
    'From' => 'richard@phpguru.org' ,
    'Subject' => 'Test mime message'
);

//宣告一個新的 Mail_mime 物件
$mime = new Mail_mime($CrLf);

//設定文字模式的內容
$mime->setTXTBody($text);

//設定 HTML 模式的內容
$mime->setHTMLBody($html);

//將檔案附上，檔案為文字形式非二次元檔。
```

```
$mime->addAttachment($file, 'text/plain');

//將所有$mime的內容放入$body中
$body = $mime->get();

//將$headers陣列設為信件的表頭
$headers = $mime->headers($headers);

//用 Mail::factory 方法產生 mail 物件
$mail =& Mail::factory('mail');
$mail->send('joe@example.com', $headers, $body);
?>
```

5-17 XML_Serializer 套件

此套件可以用 php 來建立 xml 格式的 RSS(RDF Site Summary)文件，所謂的 RSS 發展的目的是想將網站新聞以 XML 為基礎的方式來提供。它是屬於半主動的傳播方式。使用者可以設定想要那些網站主動提供的訊息，這些將會出現在瀏覽器中。例如下圖，用 FeedDemon1.0(32)這個瀏覽器當範例來接收各個網站提供的消息。



圖四十三 FeedDemon 瀏覽器外觀圖

底下利用官方網站提供的手冊中 Martin Jansen, Alexander Merz(2004)兩位所撰寫的範例程式來加以說明 XML_Serializer 套件的使用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

上圖的新聞檔的 XML 原始碼應該如下：

```

<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF>

    //提供新聞得頻道名稱
    <channel>
        <title>龍潭測試</title>
        <link>http://freedomink.org/</link>
        <item>

            //第一則新聞
    
```

```

        <item>
            <title>First Article</title>
            <link>http://freedomink.org/node/view/55</link>
            <description>Short blurb about
article.....</description>
        </item>

        // 第二則新聞
        <item>
            <title>Second Article</title>
            <link>http://freedomink.org/node/view/11</link>
            <description>This article shows you
how .....</description>
        </item>

    </item>

</channel>
</rdf:RDF>

```

要產生這樣的 xml 檔，不用會寫 xml，可以用以下的 php 檔達到同樣的效果：

```

<?php
//引入 Serializet.php 檔
require_once 'XML/Serializer.php';

//將整個文件的一些基本設定放入$options 的陣列中
$options = array(
    "indent" => "    ",
    "linebreak" => "\n",
    "typeHints" => false,

```

```

        "addDecl"    => true ,
        "encoding"   => "UTF-8" ,
        "tagName"    => "rdf:RDF" ,
        "defaultTagName" => "item"
    );

//將第一則訊息放入$stories 陣列中
$stories[] = array(
    'title'        => 'First Article' ,
    'link'         => 'http://freedomink.org/node/view/55' ,
    'description'  => 'Short blurb about article.....'
);

//將第二則訊息放入$stories 陣列中
$stories[] = array(
    'title'        => 'Second Article' ,
    'link'         => 'http://freedomink.org/node/view/11' ,
    'description'  => 'This article shows you how .....'
);

//將頻道的一些訊息放入，並將$stories 陣列放入頻道中
$data['channel'] = array(
    "title" => "龍潭測試" ,
    "link"  => "http://freedomink.org/" ,
    $stories
);

//建立 XML_Serializer 物件
$serializer = new XML_Serializer($options);

//用 XML_Serializer 物件的 serialize()方法來產生 XML 文件

```

```
if ($serializer->serialize($data)) {
    header('Content-type: text/xml');
    echo $serializer->getSerializedData();
}
?>
```

只要在 FreeDemon1.0 瀏覽器中將此網頁的網址加入到頻道中，以後就隨時可以看到由提供新聞的 Server 主動發佈的新聞，如圖四十三。

5-18 Crypt_HMAC 套件(33)

這是一個將訊息編碼的套件，我們可以用來認證使用者的身分並給予權限，由於網路上有許多人用探嗅器(sniffer)來竊取帳號密碼，若透過訊息編碼將會大大提高安全性。

底下利用 Davey Shafik (2003)所撰寫的範例程式來加以說明如何用一個秘密的 key 來為兩端的訊息的正確性作確認。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

```
<?php
//啟動 session，有了 session 變數才可跨網頁
session_start();

//若沒有$_SESSION['HMAC_key']這個 session 變數則以 md5 雜湊(hash)
//隨機產生的數字的方式來產生一組 key 值
if(!isset($_SESSION['HMAC_key'])) {
    $hmac_key = md5(uniqid(mt_rand(), 1));
    $_SESSION['HMAC_key'] = $hmac_key;
}

//將$_POST 的值傳到 server 端後，在 server 加以雜湊
if(isset($_POST['submit'])) {
```

```

require 'Crypt/HMAC.php';
$ hmac_key = $_SESSION['HMAC_key'];

//以$_SESSION['HMAC_key']來產生一個 Crypt_HMAC 的物件
$Crypt_HMAC = new Crypt_HMAC($ hmac_key, 'md5');
$user_pass = $_POST['HMAC_user'];
$user_pass .= $_POST['HMAC_pass'];

//將使用者填在表單上的「帳號」「密碼」合起來產生雜湊值，這個
//過程是在 Server 端進行
$HMAC_hash = $Crypt_HMAC->hash($user_pass);
}
?>

//網頁的 HTML 內容如下
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Strict//EN"
"http://w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>
Creating clientside digests using HMAC and MD5
</title>
//引入 HMAC_md5.js 這個檔執行 client 端的加密。
<script type="text/javascript" src="HMAC_md5.js"></script>

<script type="text/javascript">
window.HMAC_key =
'<?php
echo $_SESSION['HMAC_key'];
?>';

```

```

//建立一個 javascript 的 do_submit 函式
function do_submit() {
var username =document.getElementById(' HMAC_user' ).value;
var password =document.getElementById(' HMAC_pass' ).value;
var user_pass = username + password;

<!--用 hex_hmac 的方法將 user_pass 雜湊成 HMAC_key 的值，放在 input 元件名
為 HMAC_key 之中，在 client 端產生雜湊值-->
document.getElementById(' HMAC_hash' ).value=hex_hmac(window.HMAC_key ,
user_pass);
}

</script>
</head>
<body>
<?php
if (isset($HMAC_hash)) {
?>
<p><strong>PHP Generated Digest:</strong>
<?php echo $HMAC_hash; ?>

<br /><strong>JavaScript Generated Digest:</strong>
<?php echo $_POST[' HMAC_hash' ]; ?>
</p>

<?php
}
?>
<form method="post" action="<?php echo $_SERVER[' PHP_SELF' ]; ?>"
onsubmit="do_submit();">

```

```

<p>
<input type="text" name="HMAC_user" id="HMAC_user" />
<input type="text" name="HMAC_pass" id="HMAC_pass" />
<input type="hidden" name="HMAC_hash" id="HMAC_hash" />
<input type="submit" value="Login!" name="HMAC_submit" />
</p>
</form>
</body>
</html>

```

本範例執行的結果如下圖，填入帳號，密碼後按「Login」鈕，結果將帳號密碼合起來產生兩組編碼過的字串，「PHP Generated Digest」代表是由 Server 編碼產生的，而「JavaScript Generated Digest」代表是由 Client 端編碼產生的。發現用同一個 key 可以得到相同的編碼結果。



圖四十四 Crypt_HMAC 套件範例執行結果

5-19 HTMLArea 套件(35)

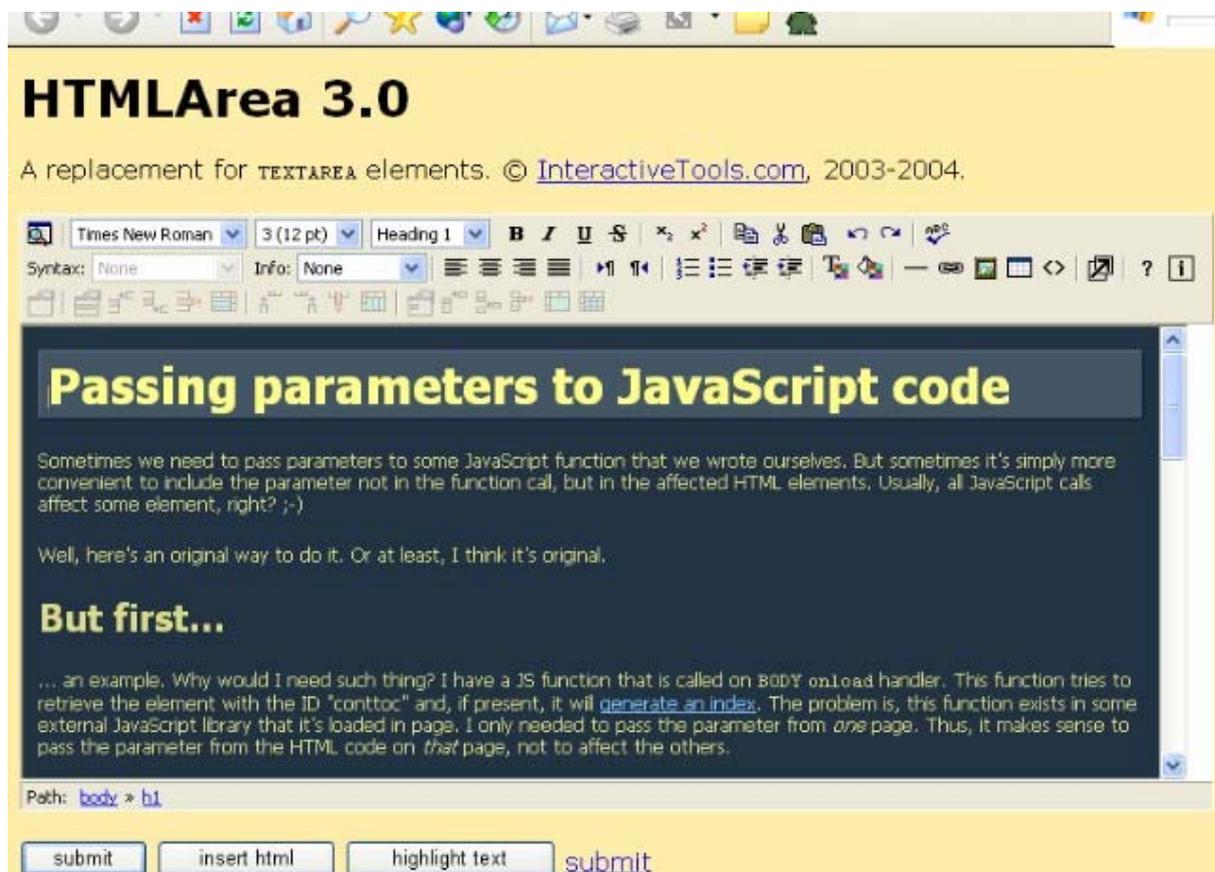
這是一個很特殊的套件，因為目前此套件尚未放入 PEAR 中，使用者必須從提供此套件的官方網站下載。官方網站網址為 <http://dynarch.com/mishoo/htmlarea.epl>

5-19.1 何謂 HTMLArea:

這是一個免費的，所見即所得的(WYSIWYG)一種運用在網頁上的編輯器，類似像 MS Word，MS FrontPage，Macromedia Dreamweaver 這種應用軟體，只是它可以在網頁上直接做編輯。這個套件是以 javascript 碼製作而成，完全由 client 端來執行，但可以搭配 PHP，Asp 等與 Server 端結合。

常見的功能有改字型，大小，顏色，靠左對齊，置中，插入圖片，觀看 HTML 原始碼…等。這個套件使用上有一些限制，IE 瀏覽器需 5.5 版以上，若是 Mozillz 瀏覽器需 1.3-Beta 版以上。

5-19.2 HTMLArea 編輯器的範例



圖四十五 HTMLArea 編輯器外觀

如圖 全能型的 HTMLArea 編輯器

HTMLArea 編輯器有各種類型，包括功能較陽春的基本型，或基本型加表格功能，基本型加檢視原始碼功能，基本型加拼字檢查功能，基本型加內容選單功能，全能型，每種類型的安裝方式與伺服器條件會有些許差異。

5-19.3 如何將 HTMLArea 基本型的編輯器加到網頁上

1. 下載最新版本的套件，目前最新版本為 HTMLArea-3.0-rc1
2. 將下載的 zip 檔解壓縮
3. 在網頁根目錄下建立一個名為 htmlarea 的目錄
4. 將解壓縮後的內容上傳到 htmlarea 目錄中

5. 打開 `htmlarea` 目錄內的範例檔看是否有上傳成功，例如
`http://YourWebSite/htmlarea/examples/core.html`
接下來要將套件內的 `javascript` 碼加到網頁中，先在網頁中定義
`_editor_url` 與 `_editor_lang`，`_editor_url` 代表絕對的 `url`，而
`_editor_lang` 代表使用的語言，可到 `/htmlarea/lang` 中去看該套件目
前支援那些語言。例如

```
<script type="text/javascript">
    _editor_url = "/htmlarea/";
    _editor_lang = "en"; //台灣可用 b5，代表 big5 碼
</script>

//引入 htmlarea.js 檔
<script type="text/javascript"
src="/htmlarea/htmlarea.js"></script>

//將<textarea>轉為 HTMLArea 的文字區塊
<script type="text/javascript" defer="1">
    HTMLArea.replaceAll();
</script>]

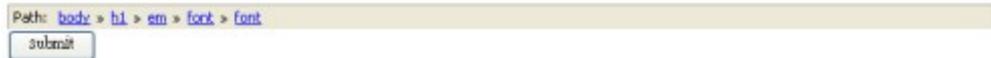
//表單的 HTML
<h1>表單範例</h1>
<form action="test.php" method="post" id="edit" name="edit">
//底下這個 TextArea 會被取代成 htmlarea 編輯器
<textarea id="ta" name="ta" style="width:100%" rows="20"
cols="80">
基本型的 HtmlArea 編輯器
</textarea>
<input type="submit" name="ok" value=" submit " />
</form>
```

上述程式碼執行結果如圖：

表單範例



基本型的HTMLArea編輯器



圖四十六 基本型的 HTMLArea 編輯器

假如一個網頁中有許多的 <textarea>，而只想讓其中一個為 HTMLArea，則 javascript 碼可改為 HTMLArea.replace(“id”)，其中的 id 為要變成編輯器的 textarea 的 id。

5-19.4 如何更改 HTMLArea 編輯器的設定

底下為範例程式碼：

```
//建立一個 HTMLArea.Config 物件，此物件擁有許多預設值
var config = new HTMLArea.Config();
//修改編輯器的寬與高
config.width = '90%';
config.height = '200px';

//修改網頁的<body>的屬性為，背景色為黃色，文字顏色為黑色，
//字型為 verdana，sans-serif，修改<p>屬性為文字為粗體
config.pageStyle = 'body { background-color: yellow; color: black;
font-family: verdana, sans-serif } ' + 'p { font-weight: bold; }
';

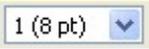
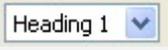
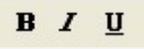
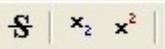
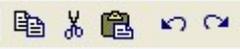
//將某一個 id 的 textarea 改為上述的設定
```

```
HTMLArea.replace('id', config);
```

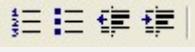
5-19.5 如何修改 HTMLArea 編輯器的工具列按鈕的設定

一樣可以用 HTMLArea.Config 物件來修改，範例程式碼如下：

```
//建立一個 HTMLArea.Config 物件，此物件擁有許多預設值
var config = new HTMLArea.Config();

//修改按鈕的設定
Config.toolbar = [
[ "fontname", "space",
//如圖：
"fontsize", "space",
//如圖：
"formatblock", "space",
//如圖：
"bold", "italic", "underline", "separator",
//如圖：
"strikethrough", "subscript", "superscript", "separator",
//如圖：
"copy", "cut", "paste", "space", "undo", "redo" ],
//如圖：
[ "justifyleft", "justifycenter", "justifyright",
"justifyfull", "separator",
//如圖：
```

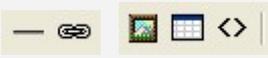
```
"insertorderedlist" , "insertunorderedlist" , "outdent" ,  
"indent" , "separator" ,
```

```
//如圖：
```

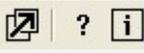
```
"forecolor" , "hilitecolor" , "separator" ,
```

```
//如圖：
```

```
"inserthorizontalrule" , "createlink" , "insertimage" ,  
"inserttable" , "htmlmode" , "separator" ,
```

```
//如圖：
```

```
"popueditor" , "separator" , "showhelp" , "about" ]
```

```
//如圖：
```

```
];
```

```
//將某一個 id 的 textarea 改為上述的設定
```

```
HTMLArea.replace('id' , config);
```

5-19.6 如何建立自製的 HTMLArea 編輯器的工具列按鈕

若要建立自製的按鈕，須先在 config.btnlist 中定義按鈕，需定義包括，按鈕名稱，圖案路徑，使用工具為何，在文字模式下是否可用，按下去後會發生的事件。名稱最好前面加上「my_」字樣，以免與其他名稱相同產生衝突。

範例程式如下：

```
//建立一個 HTMLArea.Config 物件，此物件擁有許多預設值
```

```
var config = new HTMLArea.Config();
```

```
//定義新按鈕到 config.btnlist 中。
```

```
//參數有按鈕名稱，使用工具，圖案路徑，文字模式是否可用
```

```
//作用的事件函式
```

```

config.registerButton("my-hilite", "Highlight text",
"my-hilite.gif", false, )

//當按鈕按下會啟動的事件函式，editor 是對應到 HTMLArea 物件。
function(editor, my-hilite) {
    editor.surroundHTML('<span class="hilite">', '</span>');
}
);

```

最後加此按鈕加入到工具列中，範例程式如下：

```

config.toolbar.push([ "my-hilite" ]);

```

也可用上述用 config 物件的方法，將按鈕加入到工具列中。

```

config.toolbar = [
    ['fontname', 'space',
    'fontsize', 'space',
    'formatblock', 'space',
    'separator', 'my-hilite', 'separator', 'space',
    //加入自製按鈕於此
    'bold', 'italic', 'underline', 'space']
];

```

若設定了四個按鈕，並以同一個事件函式 clickHandler 來處理，完整範例如下，

```

//先定義事件函式，用 switch 來判斷按鈕
function clickHandler(editor, buttonId) {
    switch (buttonId) {
        case "my-toc":
            editor.insertHTML("<h1>Table Of Contents</h1>");
            break;
        case "my-date":

```

```

        editor.insertHTML((new Date()).toString());
        break;
    case "my-bold":
        editor.execCommand("bold");
        editor.execCommand("italic");
        break;
    case "my-hilite":
        editor.surroundHTML("<span class=\"hilite\">" ,
"/span>");
        break;
    }
};

//建立一個 HTMLArea.Config 物件，此物件擁有許多預設值
var config = new HTMLArea.Config();

//設定四個按鈕
config.registerButton("my-toc" , "Insert TOC" , "my-toc.gif" ,
false , clickHandler);
config.registerButton("my-date" , "Insert date/time" ,
"my-date.gif" , false , clickHandler);
config.registerButton("my-bold" , "Toggle bold/italic" ,
"my-bold.gif" , false , clickHandler);
config.registerButton("my-hilite" , "Hilite selection" ,
"my-hilite.gif" , false , clickHandler);

//將按鈕放到編輯器中
config.toolbar.push(["linebreak" , "my-toc" , "my-date" ,
"my-bold" , "my-hilite"]);

//將指定的 textarea 加以取代成 HTMLArea 編輯器

```

```
HTMLArea.replace("textAreaID", config);
```

有了這個編輯器，將會讓使用者很方便的直接修改網頁內容。

PEAR 套件還有許多，研究者只針對國中化學教學網站可能會用到的套件加以探討，其他的套件，研究者期望日後建立一個共同研究的機制，例如：台灣的 PEAR 論壇。引入有同樣愛好的「仁人志士」，一起共襄盛舉。

