|  |
| --- |
| **LAB 12**  **PHP Mảng và Biến siêu toàn cục**    **Bạn sẽ học gì**  • Cách tạo, duyệt, truy cập và chèn vào mảng PHP  • Nóng để sử dụng các khóa không số làm chỉ mục trong mảng  • Cách PHP làm cho các biến HTTP dễ dàng truy cập thông qua các siêu lớp  • Cách viết một tập lệnh đáp ứng sự lựa chọn của người dùng.  **Thời gian xấp xỉ**  Các bài tập trong phòng thí nghiệm này sẽ mất khoảng 90 phút để hoàn thành. |
|  |
| Textbook by Pearson  http://www.funwebdev.com    Date Last Revised: Feb 24, 2017 |

# Mảng

P R E P A R IN G D IR E C T O R IE S

1 Nếu bạn chưa làm như vậy, hãy tạo một thư mục trong ổ đĩa cá nhân của bạn cho tất cả các phòng thí nghiệm cho cuốn sách này. Giống như Lab 11, bộ phòng thí nghiệm này yêu cầu một máy chủ web hoạt động để giải thích mã PHP. Nếu bạn không có máy chủ web đang chạy, hãy quay lại các bài tập Lab cuối cùng và có được một thiết lập.

2 Từ thư mục phòng thí nghiệm chính (được tải xuống từ trang web của sách giáo khoa bằng cách sử dụng mã được cung cấp với sách giáo khoa hoặc ở một vị trí chung được cung cấp bởi người hướng dẫn của bạn), sao chép thư mục có tiêu đề lab12 vào thư mục khóa học của bạn được tạo ở bước một.

Mảng là cấu trúc dữ liệu quan trọng trong lập trình vì chúng cho phép

lập trình viên để quản lý các bộ đối tượng như một thực thể duy nhất. Trong PHP, mảng

có tầm quan trọng bổ sung dưới dạng các mảng siêu toàn cầu chứa

dữ liệu được gửi từ máy khách đến máy chủ.

# Bắt đầu với mảng PHP

**EXERCISE 12.1** — U S E PHP A R R A Y S

1. Kiểm tra lab12-tests01.php trong trình soạn thảo văn bản và trong trình duyệt của bạn. Lưu ý rằng trang

sẽ chứa một dự báo thời tiết

2 Để xác định thời tiết trong 7 ngày tới, chúng tôi sẽ sử dụng các mảng để lưu trữ dự kiến

cao, thấp và loại thời tiết mỗi ngày.

Mặc dù sau này bạn sẽ tìm hiểu về các cấu trúc dữ liệu hiệu quả hơn để lưu trữ dữ liệu liên quan,

bây giờ chúng ta sẽ định nghĩa ba mảng riêng biệt: một cho ngày, một cho cao

nhiệt độ, một cho những người thấp. Xác định các mảng bên trong <? Php ...?>

các thẻ như sau:

**$days = array("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",**

**"Saturday", "Sunday");**

**$highs = array(20,30,26,30,30,29,25);**

**$lows = array(10,11,15,18,20,13,11);**

3 Bây giờ chúng ta sẽ xác định một hàm có 3 tham số: ngày trong tuần, mức cao

giá trị, và giá trị thấp. Lưu ý rằng đánh dấu đang được đầu ra trong chức năng này là

tương tự như trong container <div> bên dưới tiêu đề.

*/\**

*Outputs the relevant bootstrap markup to display the weather forecast for a single day*

*\*/*

**function outputForecast($day,$high,$low) {**

**echo '<div class="panel panel-default col-lg-3 col-md-3 col-sm-6">'; echo '<div class="panel-heading">'; echo '<h3 class="panel-title">' . $day . '</h3>';**

**echo '</div>';**

**echo '<div class="panel-body">'; echo '<table class="table table-hover">'; echo '<tr><td>High:</td><td>' . $high . '</td></tr>'; echo '<tr><td>Low:</td><td>' . $low . '</td></tr>'; echo '</table>'; echo '</div>'; echo '</div>';**

**}**

4 Nhắc lại rằng để truy cập các giá trị trong một mảng, bạn sử dụng ký hiệu dấu ngoặc vuông trong đó

chìa khóa được đặt trong dấu ngoặc. Thay thế đánh dấu container bằng cách sau đây

<div class="container theme-showcase" role="main">

<?php

**outputForecast($days[0], $highs[0], $lows[0]);**

?>

</div>

5 Kiểm tra trên trình duyệt.

6 Bây giờ thêm vào một vòng lặp để trang của chúng tôi hiển thị dự báo cho cả tuần

<div class="container theme-showcase" role="main">

<?php

**for ($i=0; $i<count($days); $i++) {**

outputForecast($days[**$i**], $highs[**$i**], $lows[**$i**]);

**}**

?>

</div>

*Lưu ý rằng vòng lặp này gọi hàm mới cho mỗi phần tử trong mảng $ days. Các*

*kết quả sẽ trông tương tự như kết quả trong Hình 12.1*

7 Khóa mảng trong PHP có thể là số nguyên hoặc chuỗi. Sử dụng các khóa chuỗi thường có thể tạo ra mảng của bạn

mã trực quan hơn. Sửa đổi mảng của bạn như sau

**$highs = array("Monday" => 20,"Tuesday" => 30,"Wednesday" => 26,"Thursday"**

**=> 30,"Friday" => 30,"Saturday" => 29,"Sunday" => 25);**

**$lows = array("Monday" => 10,"Tuesday" => 11,"Wednesday" => 15,"Thursday"**

**=> 18,"Friday" => 20,"Saturday" => 13,"Sunday" => 11);**

Bây giờ sửa đổi các vòng lặp của bạn như sau và sau đó kiểm tra (kết quả trong trình duyệt sẽ là

tương tự)

<div class="container theme-showcase" role="main">

<?php

*/\* version 2 \*/*

**foreach ($highs as $key => $todayHigh) { outputForecast($key, $todayHigh, $lows[$key]);**

**}**

?>

</div>

**EXERCISE 12.2** — IT E R A T IN G T H R O U G H A 2D A R R A Y

*Figure*

*12*

*.*

*1*

*–*

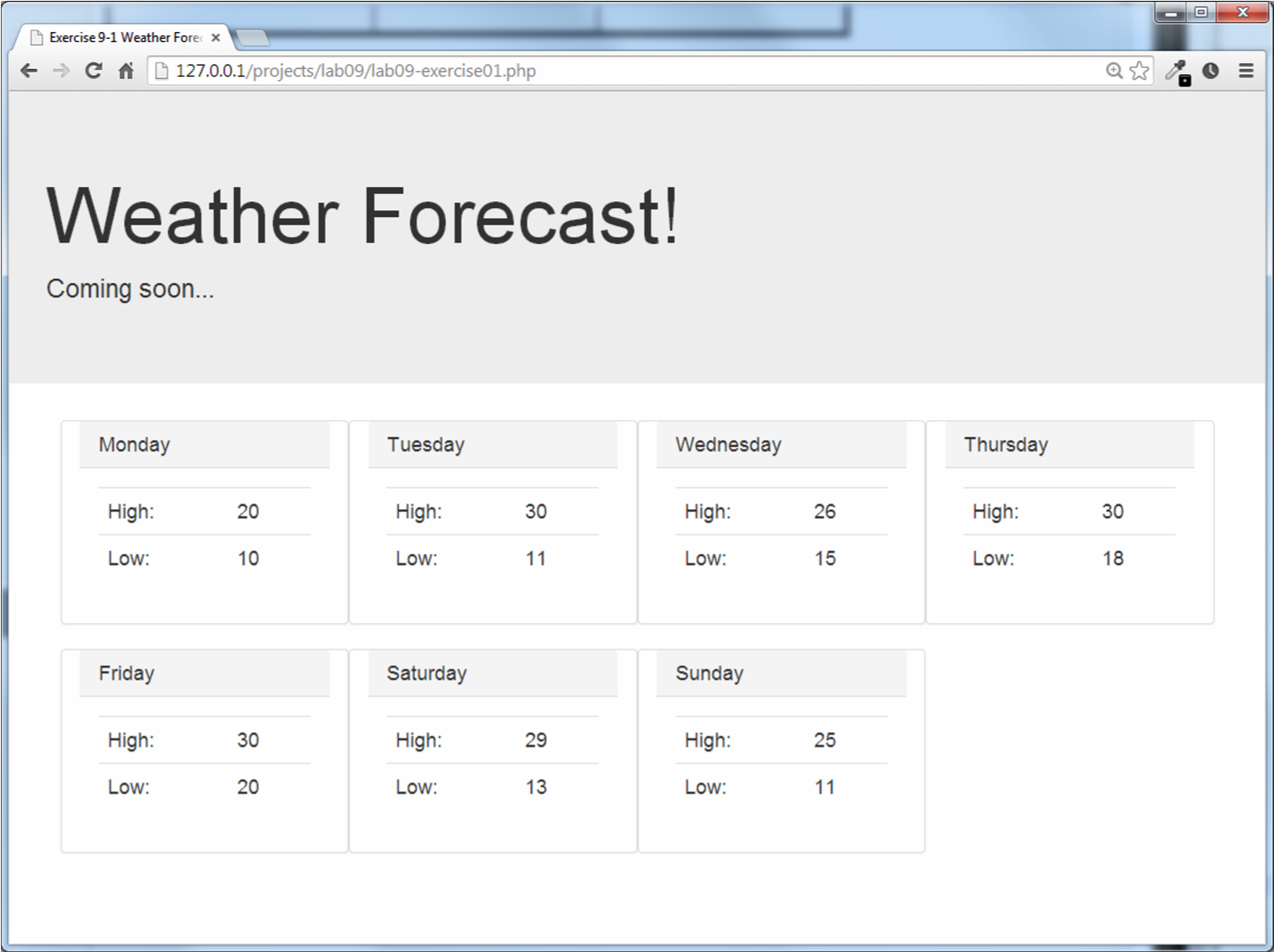
*Completed*

*E*

*xercise*

*1*

*completed.*



1 Để chứng minh việc sử dụng mảng 2D, chúng tôi sẽ cải thiện và xây dựng trên Bài tập 1 (hoặc

bạn có thể mở lab12-tests02.php ).

Trong khi trước đây chúng ta có hai mảng song song (một mức cao và một mức thấp) trước đó, chúng ta sẽ

được phục vụ tốt hơn bởi một mảng duy nhất trong đó mỗi ngày trong tuần có một mảng các giá trị thay vì

hơn một. Thay thế hai mảng thời tiết của bạn bằng cách sau:

**$forecast = array("Monday" => array(20,10),**

**"Tuesday" => array(30,11),**

**"Wednesday" => array(26,15),**

**"Thursday" => array(30,18), "Friday" => array(30,20),**

**"Saturday" => array(29,13),**

**"Sunday" => array(25,11));**

2 Vì bài tập trước đã sử dụng hai mảng, chúng ta sẽ phải thực hiện một số

thay đổi để làm cho nó hoạt động với một. Sửa đổi vòng lặp của bạn như sau

<div class="container theme-showcase" role="main">

<?php

foreach (**$forecast as $key => $weather**) { outputForecast($key, **$weather[0], $weather[1]**);

}

?>

</div>

*Lưu ý rằng so với bước 8 trong bài tập trước, mã vòng lặp của chúng ta bây giờ trông*

*sạch hơn một chút*

3 Để minh họa tại sao mảng 2d linh hoạt hơn các mảng song song bây giờ chúng ta sẽ thêm một

yếu tố thứ ba để dự báo của chúng tôi: một mô tả chuỗi.

Sửa đổi từng phần tử trong khai báo mảng dự báo $ để lấy phần tử thứ ba

$forecast = array("Monday" => array(20,10**,"Cloudy"**),

"Tuesday" => array(30,11**,"Sunny"**),

"Wednesday" => array(26,15**,"Sunny"**),

"Thursday" => array(30,18**,"Cloudy"**),

"Friday" => array(30,20**,"Rain"**),

"Saturday" => array(29,13**,"Rain"**), "Sunday" => array(25,11**,"Cloudy"**)); 4 Sửa đổi hàm outputForecast () như sau

function outputForecast($day,$high,$low**, $description**) { echo '<div class="panel panel-default col-lg-3 col-md-3 col-sm-6">'; echo '<div class="panel-heading">'; echo '<h3 class="panel-title">' . $day . '</h3>'; echo '</div>';

echo '<div class="panel-body">'; echo '<table class="table table-hover">'; echo '<tr><td>High:</td><td>' . $high . '</td></tr>'; echo '<tr><td>Low:</td><td>' . $low . '</td></tr>'; echo '</table>'; echo '</div>';  **echo '<div class="panel-footer"><img src="' . $description .**

**'.png" /> ' . $description . '</div>';**  echo '</div>';

}

5 Sửa đổi vòng lặp đầu ra như sau

foreach ($forecast as $key => $weather) {

outputForecast($key, $weather[0], $weather[1]**, $weather[2]**); }

6 Kiểm tra trên trình duyệt. Kết quả sẽ trông tương tự như trong Hình 12.2

**EXERCISE 12.3** — A R R A Y S O R T IN G

*Figure*

*12*

*.*

*2*

*Output*

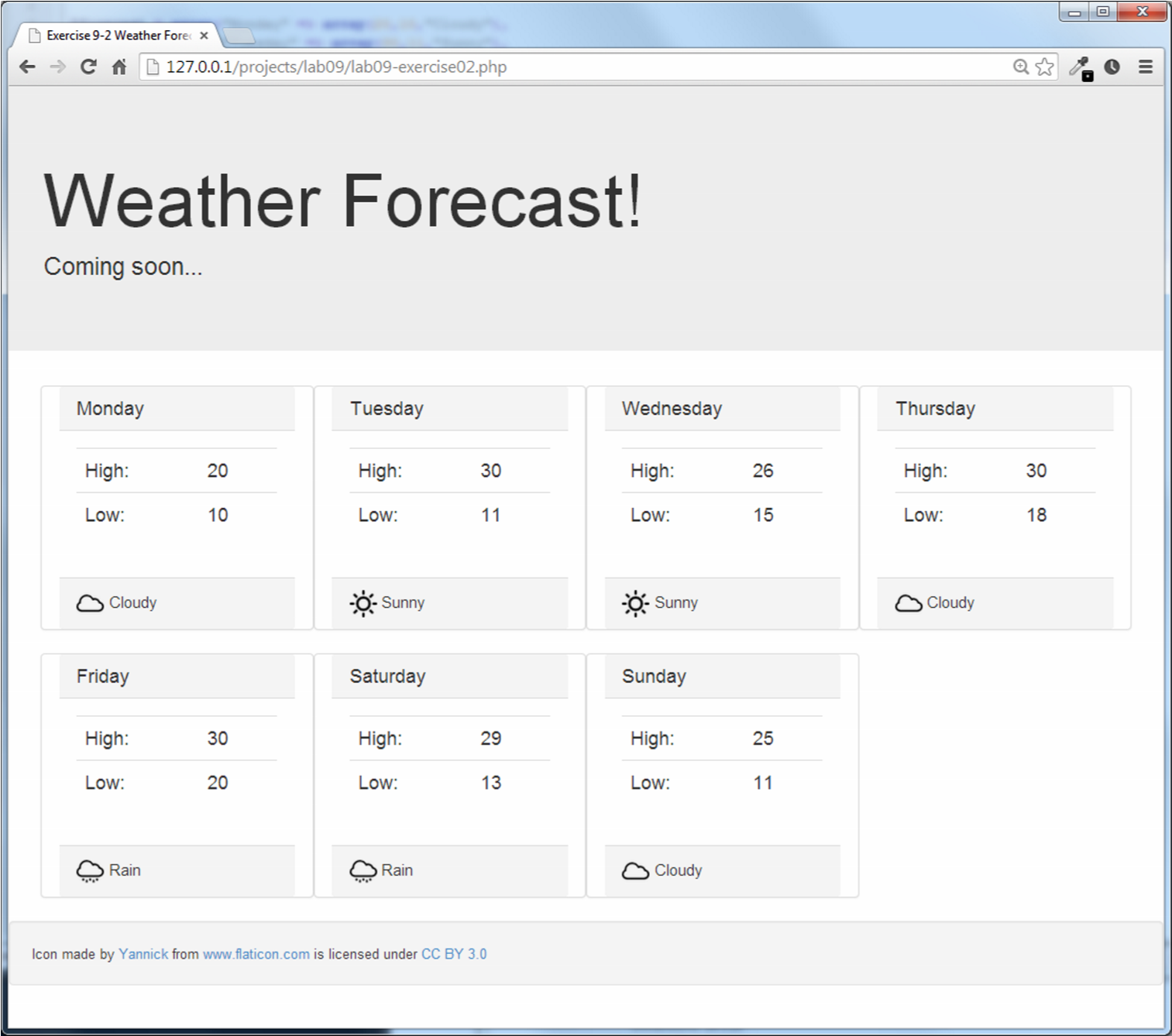
*from*

*the*

*completed*

*Exercise*

*2*



1 Mở lab12-tests03.php trong trình soạn thảo văn bản và trình duyệt của bạn. Chú ý nó in một số

bảng xếp hạng thể thao không theo thứ tự cụ thể nào sử dụng hàm print\_f () .

2 Dữ liệu được lưu trữ trong một mảng trong đó các khóa là tên của người chơi và

giá trị là điểm của họ trong mùa này. Bạn có thể sắp xếp mảng bằng hàm sort () .

Thêm các dòng sau

echo "<pre>"; print\_r($players); **sort($players); print\_r($players);** echo "</pre>";

3 Lưu và kiểm tra trang. Vì đây là một mảng kết hợp, đây không phải là kết quả chúng tôi

có lẽ muốn vì hàm sort () gán các khóa mới cho các phần tử trong

mảng. Đó là, nó sẽ xóa bất kỳ khóa hiện có nào có thể đã được chỉ định, thay vì

chỉ sắp xếp lại các phím.

3 Rõ ràng chúng ta cần giữ các khóa và giá trị liên kết với nhau. Hãy để chúng tôi thử

một chức năng sắp xếp, cụ thể là asort () chức năng

**asort**($players); print\_r($players);

*Lưu cái này và làm mới trang. Các mảng bây giờ được sắp xếp từ nhỏ nhất đến lớn nhất.*

4 Bây giờ chúng ta đảo ngược mảng đã sắp xếp sao cho điểm lớn nhất nằm trên cùng. Thêm dòng này ngay

sau khi sắp xếp

asort($players);

**$players =array\_reverse($players);**

5 Cuối cùng thay thế đầu ra print\_r () của mảng bằng một bảng được định dạng độc đáo hơn

chẳng hạn như trong Hình 12.3

5 Cuối cùng thay thế đầu ra print\_r () của mảng bằng một bảng được định dạng độc đáo hơn

chẳng hạn như trong Hình 12.3

5 Cuối cùng thay thế đầu ra print\_r () của mảng bằng một bảng được định dạng độc đáo hơn

chẳng hạn như trong Hình 12.3

. 5 Cuối cùng thay thế đầu ra print\_r () của mảng bằng một bảng được định dạng độc đáo hơn

*Figure*

*12*

*.*

*3*

*The*

*sorted*

*array*

*from*

*Exercise*

*3*

*formatted*

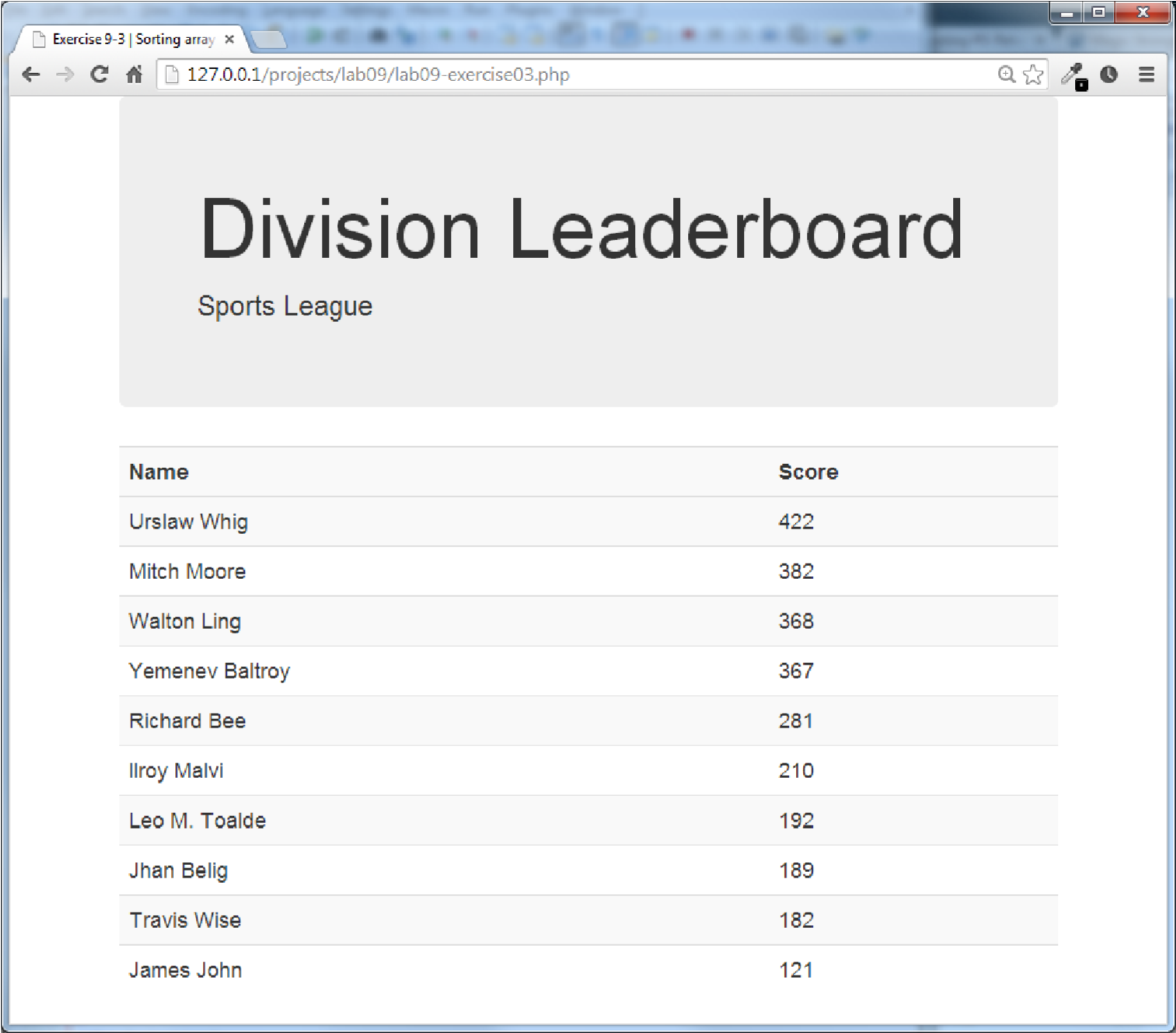
*in*

*a*

*table*

*for*

*output.*



chẳng hạn như trong Hình 12.3

**$\_GET AND $\_POST**

**EXERCISE 12.4** — C H E C K IN G F O R POST

1 Kiểm tra lab12-tests04.php trong trình duyệt và nhận thấy rằng nó chứa một trường đơn giản

hình thức gửi kết quả trở lại chính nó.

2 Điều đầu tiên để làm là phát hiện xem trang có được truy cập thông qua get hay không (một liên kết

theo dõi) hoặc liệu trang đã được gửi đến bởi biểu mẫu được đăng.

Chỉnh sửa hàm displayPostStatus () như sau và kiểm tra.

function displayPostStatus() {  **if ($\_SERVER["REQUEST\_METHOD"] == "POST") { echo "Post detected";**

**} else{**

echo "No Post Detected";

**}**

}

*Bây giờ khi bạn đăng mẫu bạn sẽ thấy bài đăng được phát hiện và khi bạn nhập URL*

*và nhấn enter (GET) nó không thấy POST.*

3 Tiếp theo, chúng ta hãy sử dụng các giá trị được truyền đến máy chủ.

Bên trong mã khi phát hiện POST, chúng ta có thể in ra toàn bộ nội dung của

Mảng $ \_POST để xem chúng ta có những biến nào khi sử dụng print\_

.

echo "<pre>"; print\_r($\_POST); echo "</pre>":

Bạn sẽ thấy rằng trường duy nhất được đăng (và có thể được nhìn thấy trong biểu mẫu HTML)

là bí danh Nhận xét các dòng này ngay bây giờ và mang chúng trở lại để giúp gỡ lỗi nếu

cần thiết

Bây giờ chúng ta sẽ nói "Xin chào người dùng", trong đó người dùng là giá trị được gửi qua trường nhập liệu

bí danh. Sửa đổi của chúng tôi trước khi tuyên bố như sau và thử nghiệm

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  **echo "Hello ". $\_POST['alias'];**

}

4 Điều này hoạt động, và sẽ nói xin chào với bất cứ điều gì bạn gõ vào hộp. Bây giờ hãy làm một chút

kiểm tra để đảm bảo nó không trống. Đặt tên tiếng vang duy nhất bên trong và sửa đổi nếu

tuyên bố để kiểm tra trống

if (isset($\_POST['alias']) **&& $\_POST['alias']!=""**) { echo "Hello ".$\_POST['alias'];

} else{

echo "Post detected, but no value"; }

5 Kiểm tra trên trình duyệt. Bây giờ trang nói xin chào nếu bạn nhập tên và người khác phát hiện bài đăng

nhưng có thể xử lý bài phù hợp nếu tên trống.

*Lưu ý: Trang dễ bị tấn công như XSS (xem Chương 18). Trước khi sử dụng*

*những kỹ thuật này trong sản xuất, hãy nhớ rằng chúng ta không bao giờ nên tin tưởng vào đầu vào của người dùng*

**EXERCISE 12.5** — U S IN G Q U E R Y S T R IN G V A L U E S

1 Mở, kiểm tra và kiểm tra lab12-tests05.php . Nó chứa một mảng dữ liệu và một

trang cơ bản với điều hướng.

2 Sửa đổi mã xuất ra menu bằng cách thay đổi URL của các liên kết để sử dụng truy vấn

dây. Trong khi hiện tại tất cả các liên kết là tất cả để # thay đổi chúng thành

foreach ($content as $key => $elements) {

echo "<li><a href='**?page=$pageCount'**>$key</a></li>";

$pageCount++;

}

Bây giờ khi bạn nhấp vào các liên kết, URL sẽ trông giống như

lab12-exercise05.php?page=1 ... lab12-exercise05.php?page=4

3 Để làm cho menu nổi bật chính xác mục hiện tại, chúng ta phải thêm kiểu CSS

đến <li> cho trang hiện tại. Nhưng làm thế nào để chúng ta biết chúng ta đang ở trang nào, khi nào

các trang đang được phục vụ bởi cùng một kịch bản?

Vì chuỗi truy vấn trong URL chứa trang = XXX, chúng tôi có thể kiểm tra $ \_GET

mảng siêu lớp cho trang hiện tại.

Trong Bootstrap, chúng ta chỉ cần thêm class = "active" vào <li> nếu trang hiện tại

trang. Sử dụng ký hiệu điều kiện cô đọng, chúng ta có thể thay đổi một dòng echo trong

lặp để xuất thuộc tính lớp nếu pageCount bằng giá trị trong mảng $ GET

foreach ($content as $key => $elements) { echo "<li ";

**echo ($pageCount==$\_GET['page'] ? " class='active'" : "");**  echo "><a href='?page=$pageCount'>$key</a></li>";

$pageCount++;

}

4 Nếu bạn thấy ví dụ trên hơi quá súc tích, bạn có thể thay thế nó bằng ví dụ sau

foreach ($content as $key => $elements) { echo "<li ";

**if ($pageCount==$\_GET['page']) echo " class='active'"; else echo "";**

echo "><a href='?page=$pageCount'>$key</a></li>";

$pageCount++;

}

5 Bây giờ chúng ta hãy sử dụng mảng dữ liệu để xuất tiêu đề và nội dung trong trang.

Trong phần đầu của trang, hãy sửa đổi thẻ <title> để chúng tôi truy cập vào $ \_GET

mảng và sử dụng giá trị trang làm chỉ mục vào mảng nội dung $ của chúng tôi . Để làm điều này, thêm

dòng sau

$content = array("Home" => array("Home Page","This page is the home page"),

"Page 2" => array("The second page", "second page content here"),

"Page 3" => array("The third page", "third page content here"),

"Page 4" => array("The fourth page", "fourth page content here"));

**$justContent = array\_values($content);**

6 Sửa đổi phần tử <title> như sau

<title>**<?php echo $justContent[$\_GET['page']-1][0]; ?>**</title>

7 Cuối cùng, theo cách tương tự với tiêu đề, chúng tôi sẽ xuất trường nội dung cuối cùng dưới dạng nội dung trong

trang. Trang cuối cùng của bạn sẽ trông giống như trong Hình 12.4

.

*Figure*

*12*

*.*

*4*

*–*

*A*

*screen*

*shot*

*of*

*one*

*of*

*the*

*four*

*pages,*

*with*

*the*

*menu*

*highlighted*



**$\_SERVER ARRAY**

**EXERCISE 12.6** — U S IN G T H E $\_SERVER S U P E R G LO B A L

1 Kiểm tra lab12-tests06.php và sau đó kiểm tra trong trình duyệt và trình soạn thảo văn bản. bên trong

trình duyệt bạn sẽ thấy rằng tất cả các giá trị trong mảng $ \_SERVER đang được hiển thị.

Một số giá trị này được gửi bởi khách hàng (bạn tại máy tính của bạn) và các tiêu đề khác

là những phản hồi từ máy chủ.

2 Xóa lời gọi print\_r () và thay thế nó bằng hai phần làm nổi bật một số

các biến thú vị ở cả phía máy khách và máy chủ.

**echo "<h1>Server Side Values</h1>";**

**echo "SERVER\_NAME: ".$\_SERVER["SERVER\_NAME"] . "<br/>"; echo "SERVER\_SOFTWARE: ".$\_SERVER["SERVER\_SOFTWARE"] . "<br/>"; echo "SERVER\_ADDR: ".$\_SERVER["SERVER\_ADDR"] . "<br/>";**

**echo "<h1>Headers from the client</h1>"; echo "REMOTE\_ADDR: ".$\_SERVER["REMOTE\_ADDR"] . "<br/>"; echo "HTTP\_USER\_AGENT: ".$\_SERVER['HTTP\_USER\_AGENT']. "<br/>"; echo "HTTP\_REFERER: ".$\_SERVER['HTTP\_REFERER']. "<br/>";**

3 Bây giờ, hãy sử dụng chức năng tích hợp để có được thông tin dễ sử dụng hơn từ người dùng-

tiêu đề đại lý:

**$browser = get\_browser($\_SERVER['HTTP\_USER\_AGENT'], true); print\_r($browser);**

*Lưu ý: Để get\_browser () hoạt động, tệp php.ini của bạn phải trỏ đến phần lông mày*

*cài đặt đúng vị trí của tệp lông mày trên hệ thống của bạn. Một hiện tại*

*Có thể tải xuống tập tin lông mày từ php.net*

**$\_FILES Mảng**

**EXERCISE 12.7** — P R O C E S S IN G F IL E U P L O A D S

1 Mở lab12-tests07.php trong trình duyệt. Bạn sẽ nhận thấy rằng nó là một hình thức để tải lên

các tập tin. Bạn có thể thử tải lên một tệp ngay bây giờ và biểu mẫu sẽ đăng, nhưng bây giờ sẽ không có gì

xảy ra với các tập tin.

2 Để xử lý đúng tệp đã tải lên, chúng ta phải lấy tệp đã đăng và lưu trữ

vĩnh viễn trên máy chủ. Chúng tôi cũng phải cập nhật giao diện người dùng để người dùng biết

việc tải lên đã thành công.

Chúng tôi sẽ bắt đầu bằng cách xác định xem có hiển thị biểu mẫu hay thay vào đó để xử lý

đã tải lên tệp và xuất thông báo bằng cách thay đổi PHP như sau:

*// if the form was posted, process the upload* **if ($\_SERVER["REQUEST\_METHOD"] == "POST") { echo "File was sent to server <a href=''>Upload another file</a>";**

**} else {**

echo getFileUploadForm();

**}**

3 Bây giờ chúng tôi thực sự sẽ sao chép tệp đã tải lên vào vị trí chúng tôi kiểm soát.

Tạo một thư mục trong thư mục làm việc của bạn có tên UPLOADS và cấp cho người dùng apache

quyền tải lên (nếu bạn đang làm việc cục bộ bằng cách sử dụng một cái gì đó như easyPHP hoặc XAMP

đây sẽ là trường hợp) Bây giờ thêm mã vào câu lệnh if di chuyển tệp:

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  **$fileToMove = $\_FILES['file1']['tmp\_name'];**

**$destination = "./UPLOADS/" . $\_FILES["file1"]["name"]; if (move\_uploaded\_file($fileToMove,$destination)) {**

echo "File was sent to server <a href=''>Upload another file</a>";

**} else {**

**echo "there was a problem moving the file";**

**}** }

Hãy thử tải lên tệp và xác nhận rằng tệp đã được đặt trong thư mục UPLOADS

**EXERCISE 12.8** — M A N A G IN G U P L O A D E D F IL E S

1 Bài tập cuối cùng cho phép chúng tôi tải một tập tin lên một thư mục tải lên. Bài tập này

xây dựng trên đó và thêm nhiều tệp tải lên và quản lý tệp được cải thiện

đặc trưng.

2 Để làm cho mã của chúng tôi sạch hơn một chút, hãy tạo một chức năng ở đầu trang của bạn như sau.

*/\**

*Moves an uploaded file to our destination location*

*\*/* **function moveFile($fileToMove, $destination) { echo $destination . ' Uploaded successfully<br/>'; move\_uploaded\_file($fileToMove, "UPLOADS/" . $destination) or die("error");**

**}**

3 Bây giờ, bạn sẽ kích hoạt nhiều tệp tải lên. Trong trường HTML cho đầu vào tệp, bạn sẽ

cần thêm nhiều thuộc tính vào hình thức tải lên như sau (nghĩa là bạn sẽ cần

để thực hiện thay đổi đối với đánh dấu được tạo bởi hàm getFileUploadForm () )

<input type="file" name="file1" id="file1" **multiple**/>

Bây giờ nếu bạn làm mới trang và nhấp vào duyệt, bạn sẽ có thể ctrl hoặc shift shift để chọn

nhiều tập tin.

Một thay đổi khác là tên của thuộc tính phải bao gồm dấu ngoặc vuông sao cho

nó có thể được xử lý như một mảng trong PHP. Để làm như vậy, thực hiện thay đổi sau đây

.

<input type="file" **name="file1[]"** id="file1" multiple/>

4 Hiện tại khi bạn cố tải lên hai tệp, chỉ một tệp sẽ được tải lên thành công. Đến

xử lý nhiều tệp bạn phải thay đổi mã xử lý tải lên để xử lý các mảng

các giá trị (nghĩa là nhiều tệp được tải lên)

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

*// check if user uploaded multiple files*  **if (is\_array($\_FILES["file1"]["error"])) {**

**$count=0;**

*// loop through each uploaded file*  **foreach ($\_FILES["file1"]["error"] as $error) { if ($error == UPLOAD\_ERR\_OK) {**

**$clientName = $\_FILES["file1"]["name"][$count]; $serverName = $\_FILES["file1"]["tmp\_name"][$count]; moveFile($serverName, $clientName);**

**$count++;**

**}**

**} } else {**

*// user only uploaded a single file*

**if ($\_FILES["file1"]["error"] == UPLOAD\_ERR\_OK) { $clientName = $\_FILES["file1"]["name"]; $serverName = $\_FILES["file1"]["tmp\_name"]; moveFile($serverName, $clientName);**

**}**

**}** } else {

echo getFileUploadForm();

}

5 Kiểm tra trong trình duyệt và xác minh rằng tải lên và di chuyển hoạt động như mong đợi.

6 Cuối cùng, để minh họa một số xác nhận chúng ta có thể thực hiện ở phía máy chủ, hãy thêm một

hạn chế chỉ cho phép tải lên các tệp jpg và png. Xác định một số phần mở rộng hợp lệ

và các loại MIME:

function moveFile($fileToMove, $destination) {  **$validExt = array("jpg", "png");**

**$validMime = array("image/jpeg","image/png");**

7 Bây giờ trong moveFile của bạn (hàm 0, bạn sẽ cần thêm mã để kiểm tra tiện ích mở rộng và

loại mime chống lại danh sách các loại hợp lệ của chúng tôi. Bạn cũng sẽ phải vượt qua trong loại tệp của

tập tin đang được tải lên

function moveFile($fileToMove, $destination**, $fileType**) {

$validExt = array("jpg", "png");

$validMime = array("image/jpeg","image/png");

*// make an array of two elements, first=filename before extension,*

*// and the second=extension*

**$components = explode(".", $destination);**

*// retrieve just the end component (i.e., the extension)*

**$extension = end($components);**

*// check to see if file type is a valid one*

**if (in\_array($fileType,$validMime) && in\_array($extension, $validExt)) {**  echo $destination . 'Uploaded successfully<br/>'; move\_uploaded\_file($fileToMove, "UPLOADS/" . $destination) or die("error");  **} else echo 'Invalid file type=' . $fileType . ' Extension=' . $extension .**

**'<br/>';**

}

8 Thay đổi lời gọi của hàm bằng cách chuyển loại tệp.

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  *// check if user uploaded multiple files*  if (is\_array($\_FILES["file1"]["error"])) {

$count=0;

*// loop through each uploaded file*  foreach ($\_FILES["file1"]["error"] as $error) { if ($error == UPLOAD\_ERR\_OK) { $clientName = $\_FILES["file1"]["name"][$count];

$serverName = $\_FILES["file1"]["tmp\_name"][$count];  **$fileType = $\_FILES["file1"]["type"][$count];**  moveFile($serverName, $clientName**, $fileType**);

$count++;

}

} } else {

*// user only uploaded a single file*

if ($\_FILES["file1"]["error"] == UPLOAD\_ERR\_OK) { $clientName = $\_FILES["file1"]["name"];

$serverName = $\_FILES["file1"]["tmp\_name"];  **$fileType = $\_FILES["file1"]["type"];**  moveFile($serverName, $clientName**, $fileType**);

}

}

}

9 Kiểm tra trên trình duyệt. Hãy thử tải lên cả hai tệp hợp lệ (JPG hoặc PNG) và không hợp lệ .

# READING/WRITING FILES

**EXERCISE 12.9** — PHP F IL E A C C E S S

1 Cuối cùng, chúng tôi sẽ chứng minh bạn có thể đọc và ghi tệp dễ dàng như thế nào trong PHP. Kiểm tra

tệp văn bản datafile.txt trong một trình soạn thảo.

2 Mở lab12-tests09.php trong trình chỉnh sửa yêu thích của bạn.

3 Tệp này đọc dữ liệu từ một tệp và lặp lại bằng cách sử dụng câu lệnh echo .

Xóa câu lệnh echo và viết đoạn mã sau sẽ chuyển đổi chuỗi

từ tập tin thành một mảng. Chúng tôi đang sử dụng hàm explode () để tận dụng lợi thế

thực tế là tập tin này được phân định dòng mới. Chúng ta có thể xuất mảng bằng print\_r () ..

**$daysOfWeek = explode("\n",$fileContents);**

4 Bây giờ chúng ta có thể lặp qua các phần tử của mảng và từng phần tử trong mảng

tương ứng với một dòng của tập tin! Lưu ý rằng mỗi dòng của tệp là một bó số

cách nhau bằng dấu phẩy. Các giá trị là:

Ngày, tháng, năm, nhiệt độ cao, nhiệt độ thấp.

Do đó, chúng ta có thể lặp qua mảng của mình và sau đó tách từng chuỗi thành một chuỗi khác

mảng như sau

**foreach($daysOfWeek as $day) {**  $elements = explode(",",$day); echo $elements[1]; *// example* //...

1. 5 Sử dụng hàm outputForecast () từ EXERCISE 12.2, chuyển đổi dữ liệu trong tệp
2. vào đầu ra giống với dự báo 14 ngày như trong Hình 12.5

5 Sử dụng hàm outputForecast () từ EXERCISE 12.2, chuyển đổi dữ liệu trong tệpvào đầu ra giống với dự báo 14 ngày như trong Hình 12.5

*Figure*

*12*

*.*

*5*

*Illustrating*

*the*

*output*

*of*

*the*

*completed*

*EXERCISE*

*12*

