

PHP

Part 4

Topics

- The PHP date() function
- PHP include and require Statements
- PHP File Open/Read/Close/ Write
- Create an Upload-File Form
- PHP Cookies
- PHP Session Variables

The PHP Date() Function

- The PHP date() function formats a timestamp to a more readable date and time.

Syntax

```
date(format,timestamp)
```

Parameter	Description
format	Required. Specifies the format of the timestamp
timestamp	Optional. Specifies a timestamp. Default is the current da

Get a Simple Date

- Get a Simple Date
- The required *format* parameter of the date() function specifies how to format the date (or time).
- Here are some characters that are commonly used for dates:
 - d - Represents the day of the month (01 to 31)
 - m - Represents a month (01 to 12)
 - Y - Represents a year (in four digits)
 - 1 - Represents the day of the week
- Other characters, like "/", ".", or "-" can also be inserted between the characters to add additional formatting.

The PHP Date() Function

Example

```
<?php  
echo "Today is " . date("Y/m/d") . "<br>";  
echo "Today is " . date("Y.m.d") . "<br>";  
echo "Today is " . date("Y-m-d") . "<br>";  
echo "Today is " . date("l");  
?>
```

Today is 2014/08/03

Today is 2014.08.03

Today is 2014-08-03

Today is Sunday

Get a Simple Time

- Here are some characters that is commonly used for times:
 - h - 12-hour format of an hour with leading zeros (01 to 12)
 - i - Minutes with leading zeros (00 to 59)
 - s - Seconds with leading zeros (00 to 59)
 - a - Lowercase Ante meridiem and Post meridiem (am or pm)

Get a Simple Time

Example

```
<?php  
echo "The time is " . date("h:i:sa");  
?>
```

The time is 09:28:33am

DATE LAB #1

- จงเขียนโปรแกรมแสดง วัน เดือน พ.ศ. เวลา เป็นภาษาไทย

วัน จันทร์ ที่ 4 สิงหาคม พ.ศ. 2522 เวลา 17 นาฬิกา 35 นาที

PHP include and require Statements

- It is possible to insert the content of one PHP file into another PHP file
- require will produce a fatal error (E_COMPILE_ERROR) and stop the script
- include will only produce a warning (E_WARNING) and the script will continue

PHP include and require Statements

Syntax

```
include 'filename';
```

or

```
require 'filename';
```

Example PHP include

Example

Assume we have a file called "vars.php", with some variables defined:

```
<?php  
$color='red';  
$car='BMW';  
?>
```

Then, if we include the "vars.php" file, the variables can be used in the calling file:

Example

```
<html>  
<body>  
  
<h1>Welcome to my home page!</h1>  
<?php include 'vars.php';  
echo "I have a $color $car.";  
?>  
  
</body>  
</html>
```

PHP File Open/Read/Close

- How to open, read, and close a file on the server.

PHP Open File - fopen()

- A better method to open files is with the `fopen()` function.
- This function gives you more options than the `readfile()` function.

PHP Open File - fopen()

Modes	Description
r	Open a file for read only. File pointer starts at the beginning of the file
w	Open a file for write only. Erases the file if it exists. File pointer starts at the beginning of the file
a	Open a file for write only. The existing file is not erased. Creates a new file if the file doesn't exist. File pointer starts at the end of the file
x	Creates a new file for write only. Returns an error if the file already exists. File pointer starts at the beginning of the file
r+	Open a file for read/write. File pointer starts at the beginning of the file
w+	Open a file for read/write. Erases the file if it exists. File pointer starts at the beginning of the file
a+	Open a file for read/write. The existing file is not erased. Creates a new file if the file doesn't exist. File pointer starts at the end of the file
x+	Creates a new file for read/write. Returns an error if the file already exists. File pointer starts at the beginning of the file

PHP Close File - fclose()

- The fclose() function is used to close an open file.

```
<?php
$myfile = fopen("webdictionary.txt", "r");
// some code to be executed....
fclose($myfile);
?>
```

PHP Read File - fread()

- The fread() function reads from an open file.

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("webdictionary.txt"));
fclose($myfile);
?>
```


PHP Read Single Line - fgets()

- The fgets() function is used to read a single line from a file.

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fgets($myfile);
fclose($myfile);
?>
```

PHP Check End-Of-File - feof()

- The feof() function checks if the "end-of-file" (EOF) has been reached.
- The feof() function is useful for looping through data of unknown length.

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
    echo fgets($myfile) . "<br>";
}
fclose($myfile);
?>
```

PHP Read Single Character - fgetc()

- The fgetc() function is used to read a single character from a file.

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
// Output one character until end-of-file
while(!feof($myfile)) {
    echo fgetc($myfile);
}
fclose($myfile);
?>
```

PHP Create File - fopen()

- If you use fopen() on a file that does not exist, it will create it, given that the file is opened for writing (w) or appending (a)

Example

```
$myfile = fopen("testfile.txt", "w")
```

PHP Write to File - fwrite()

- The fwrite() function is used to write to a file.

Example

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

File Lab #2

- เขียน Form เก็บข้อมูล ชื่อนักศึกษา และ รหัสนักศึกษา ไว้ที่ File: studentdata.txt

Lab_file.php

Name :

Student ID :

submit

แสดงรายชื่อทั้งหมด

นาย XXXX YYYYY รหัส ZZZZZZ

นางสาว XXXX YYYY รหัส ZZZZZZ

Create an Upload-File Form

- To allow users to upload files from a form can be very useful.

```
<html>
<body>

<form action="upload_file.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file"><br>
<input type="submit" name="submit" value="Submit">
</form>

</body>
</html>
```

Create The Upload Script

```
<?php
if ($_FILES["file"]["error"] > 0) {
    echo "Error: " . $_FILES["file"]["error"] . "<br>";
} else {
    echo "Upload: " . $_FILES["file"]["name"] . "<br>";
    echo "Type: " . $_FILES["file"]["type"] . "<br>";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " kB<br>";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
?>
```


Create The Upload Script

- `$_FILES["file"]["name"]` - the name of the uploaded file
- `$_FILES["file"]["type"]` - the type of the uploaded file
- `$_FILES["file"]["size"]` - the size in bytes of the uploaded file
- `$_FILES["file"]["tmp_name"]` - the name of the temporary copy of the file stored on the server
- `$_FILES["file"]["error"]` - the error code resulting from the file upload

Saving the Uploaded File

```
<?php
$temp = explode(".", $_FILES["file"]["name"]);
$extension = end($temp);

if (file_exists("upload/" . $_FILES["file"]["name"])) {
    echo $_FILES["file"]["name"] . " already exists. ";
} else {
    move_uploaded_file($_FILES["file"]["tmp_name"],
        "upload/" . $_FILES["file"]["name"]);
    echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}

?>
```

PHP Cookies

- A cookie is often used to identify a user.
- A cookie is a small file that the server embeds on the user's computer.
- Each time the same computer requests a page with a browser, it will send the cookie too.
- With PHP, you can both create and retrieve cookie values.

Create a Cookie

- The `setcookie()` function is used to set a cookie.

Syntax

```
setcookie(name, value, expire, path, domain);
```

Create a Cookie

```
<?php  
setcookie("user", "Alex Porter", time()+3600);  
?>
```

Retrieve a Cookie Value

```
<?php
// Print a cookie
echo $_COOKIE["user"];

// A way to view all cookies
print_r($_COOKIE);
?>
```

Retrieve a Cookie Value

```
<html>
<body>

<?php
if (isset($_COOKIE["user"]))
    echo "Welcome " . $_COOKIE["user"] . "!<br>";
else
    echo "Welcome guest!<br>";
?>

</body>
</html>
```

Delete a Cookie

- When deleting a cookie you should assure that the expiration date is in the past

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time()-3600);
?>
```

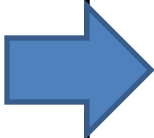

Cookie LAB #4

- ทำหน้า **Log in** สำหรับตรวจสอบว่า **username password** ถูกต้อง ให้เก็บชื่อไว้ใน **cookie : username** และแสดงหลังจาก **log in** ว่าใครเป็นคน **login**

Lab_cookie.php

Name :

Password :



ล็อกอินถูกต้อง
ผู้ใช้งาน ชื่อ : XXXXXXXXX

PHP Session Variables

- A PHP session solves this problem by allowing you to store user information on the server for later use
- session information is temporary and will be deleted after the user has left the website
- Sessions work by creating a unique id (UID) for each visitor

Starting a PHP Session

- Before you can store user information in your PHP session, you must first start up the session

```
<?php session_start(); ?>
```

```
<html>
```

```
<body>
```

```
</body>
```

```
</html>
```

Storing a Session Variable

- The correct way to store and retrieve session variables is to use the PHP `$_SESSION` variable

```
<?php
session_start();
// store session data
$_SESSION['views']=1;
?>

<html>
<body>

<?php
//retrieve session data
echo "Pageviews=". $_SESSION['views'];
?>

</body>
</html>
```

Destroying a Session

- The unset() function is used to free the specified session variable

```
<?php
session_start();
if(isset($_SESSION['views']))
    unset($_SESSION['views']);
?>
```

- destroy the session by calling the session_destroy() function

```
<?php
session_destroy();
?>
```

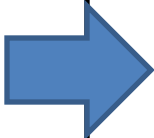
Session LAB#5

- ทำหน้า **Log in** สำหรับตรวจสอบว่า **username password** ถูกต้อง ให้เก็บชื่อไว้ใน **session : username** และแสดงหลังจาก **log in** ว่าใครเป็นคน **login**

Lab_session.php

Name :

Password :



ล็อกอินถูกต้อง
ผู้ใช้งาน ชื่อ : XXXXXXXXX