

1 PHP include file

CSC443: Web Programming

include() example

```
<a href="/default.php">Home</a>
<a href="/tutorials.php">Tutorials</a>
<a href="/references.php">References</a>
<a href="/examples.php">Examples</a>
<a href="/contact.php">Contact Us</a>
PHP

<html>
<body>

<div class="leftmenu">
<?php include("menu.php"); ?>
</div>

<h1>Welcome to my home page.</h1>
<p>I have a great menu here.</p>

</body>
</html>
PHP
```

CSC443: Web Programming

PHP Include File

2

- Insert the content of one PHP file into another PHP file before the server executes it
- Use the
 - ▣ `include()` generates a warning, but the script will continue execution
 - ▣ `require()` generates a fatal error, and the script will stop

CSC443: Web Programming

4 PHP File Input/Output

CSC443: Web Programming

PHP file I/O functions

5

function name(s)	category
file , file_get_contents , file_put_contents	reading/writing entire files
basename , file_exists , filesize , fileperms , filemtime , is_dir , is_readable , is_writable , disk_free_space	asking for information
copy , rename , unlink , chmod , chgrp , chown , mkdir , rmdir	manipulating files and directories
glob , scandir	reading directories

CSC443: Web Programming

Reading/writing files

6

contents of foo.txt	file("foo.txt")	file_get_contents("foo.txt")
Hello how are you? I'm fine	array("Hello\n", #0 "how are\n", #1 "you?\n", #2 "\n", #3 "I'm fine\n" #4)	"Hello\n how are\n you?\n \n I'm fine\n"

- `file` returns lines of a file as an array
- `file_get_contents` returns entire contents of a file as a string

CSC443: Web Programming

Reading/writing an entire file

7

```
# reverse a file
$text = file_get_contents("poem.txt");
$text = strrev($text);
file_put_contents("poem.txt", $text);
```

PHP

- `file_get_contents` returns entire contents of a file as a string
- `file_put_contents` writes a string into a file, replacing any prior contents

CSC443: Web Programming

Appending to a file

8

```
# add a line to a file
$new_text = "P.S. ILY, GTG TTYL!~";
file_put_contents("poem.txt", $new_text, FILE_APPEND);
```

PHP

old contents

Roses are red,
Violets are blue.
All my base,
Are belong to you.

new contents

Roses are red,
Violets are blue.
All my base,
Are belong to you.
P.S. ILY, GTG TTYL!~

CSC443: Web Programming

The file function

9

```
# display lines of file as a bulleted list
$lines = file("todolist.txt");
foreach ($lines as $line) {
    ?>
    <li> <?= $line ?> </li>
<?php
}
?>
```

PHP

- file returns the lines of a file as an array of strings
 - each string ends with \n
 - to strip the \n off each line, use optional second parameter:

```
$lines = file("todolist.txt", FILE_IGNORE_NEW_LINES);
```

PHP

CSC443: Web Programming

Fixed-length files, file and list

11

```
Xenia Mountroudou
(919) 685-2181
570-86-7326
```

contents of file personal.txt

```
list($name, $phone, $ssn) = file("personal.txt");
```

PHP

- reads the file into an array of lines and unpacks the lines into variables
- Need to know a file's exact length/format

CSC443: Web Programming

Unpacking an array: list

10

```
list($var1, ..., $varN) = array;
```

PHP

```
$values = array("mundruid", "18", "f", "96");
...
list($username, $age, $gender, $iq) = $values;
```

PHP

- the list function accepts a comma-separated list of variable names as parameters
- use this to quickly "unpack" an array's contents into several variables

CSC443: Web Programming

Splitting/joining strings

12

```
$array = explode(delimiter, string);
$string = implode(delimiter, array);
```

PHP

```
$class = "CS 380 01";
$class1 = explode(" ", $class); # ("CS", "380", "01")
$class2 = implode("...", $class1); # "CSE...380...01"
```

PHP

- explode and implode convert between strings and arrays

CSC443: Web Programming

Example explode

13

```
Harry Potter, J.K. Rowling  
The Lord of the Rings, J.R.R. Tolkien  
Dune, Frank Herbert
```

contents of input file books.txt

```
<?php foreach (file("books.txt") as $book) {  
    list($title, $author) = explode("", $book);  
    ?>  
    <p> Book title: <?= $title ?>, Author: <?= $author ?>  
</p>  
<?php  
}  
?>
```

PHP

CSC443: Web Programming

Reading directories

14

function	description
scandir	returns an array of all file names in a given directory (returns just the file names, such as "myfile.txt")
glob	returns an array of all file names that match a given pattern (returns a file path and name, such as "foo/bar/myfile.txt")

CSC443: Web Programming

Reading directories (Cont.)

15

```
<?php  
foreach (glob("*.txt") as $filename) {  
    echo "$filename size " . filesize($filename) . "\n";  
}  
?>
```

CSC443: Web Programming

Example for glob

16

```
# reverse all poems in the poetry directory  
$poems = glob("poetry/poem*.dat");  
foreach ($poems as $poemfile) {  
    $text = file_get_contents($poemfile);  
    file_put_contents($poemfile, strrev($text));  
    print "I just reversed " .  
basename($poemfile) ;  
}
```

PHP

- glob can match a "wildcard" path with the * character
- the basename function strips any leading directory from a file path

CSC443: Web Programming

Example for glob

17

```
# reverse all poems in the poetry directory
$poems = glob("poetry/poem*.dat");
foreach ($poems as $poemfile) {
    $text = file_get_contents($poemfile);
    file_put_contents($poemfile, strrev($text));
    print "I just reversed " .
    basename($poemfile) ;
}
```

PHP

- ❑ glob can match a "wildcard" path with the * character
- ❑ the basename function strips any leading directory from a file path

CSC443: Web Programming

Example for scandir

18

```
<ul>
<?php
$folder = "taxes/old";
foreach (scandir($folder) as $filename) {
    ?>
    <li> <?= $filename ?> </li>
<?php
}
?>
</ul>
```

PHP

```
• .
• ..
• 2009_w2.pdf
• 2007_1099.doc
```

output

CSC443: Web Programming

19

PHP Exceptions

CSC443: Web Programming

Exceptions

20

- ❑ Used to change the normal flow of the code execution if a specified error (exceptional) condition occurs.
- ❑ What normally happens when an exception is triggered:
 - ❑ current code state is saved
 - ❑ code execution will switch to a predefined (custom) exception handler function
 - ❑ the handler may then
 - resume the execution from the saved code state,
 - terminate the script execution or
 - continue the script from a different location in the code

CSC443: Web Programming

Exception example

21

```
<?php
//create function with an exception
function checkStr($str)
{
    if(strcmp($str, "correct") != 0)
    {
        throw new Exception("String is not correct!");
    }
    return true;
}

//trigger exception
checkStr("wrong");
?>
```

PHP

CSC443: Web Programming

Exception example (cont.)

22

```
<?php
//create function with an exception
function checkStr($str)
{
    ...
}

//trigger exception in a "try" block
try
{
    checkStr("wrong");
    //If the exception is thrown, this text will not be shown
    echo 'If you see this, the string is correct';
}

//catch exception
catch(Exception $e)
{
    echo 'Message: ' . $e->getMessage();
}
?>
```

PHP

CSC443: Web Programming

PHP larger example

23

- Display a random quote of the day:
- I don't know half of you half as well as I should like; and I like less than half of you half as well as you deserve.
J. R. R. Tolkien (1892 - 1973), *The Fellowship of the Ring*
- I have not failed. I've just found 10,000 ways that won't work.
Thomas A. Edison (1847 - 1931), (attributed)
- I am among those who think that science has great beauty. A scientist in his laboratory is not only a technician: he is also a child placed before natural phenomena which impress him like a fairy tale.
Marie Curie (1867 - 1934)
- I love deadlines. I like the whooshing sound they make as they fly by. **Douglas Adams**
- Statistics: The only science that enables different experts using the same figures to draw different conclusions.
Evan Esar

CSC443: Web Programming

24

PHP cookies and sessions

CSC443: Web Programming

Cookies

25

- Problem: HTTP is stateless
- What is a cookie?
 - ▣ tiny bits of information that a web site could store on the client's machine
 - ▣ they are sent back to the web site each time a new page is requested by this client.



CSC443: Web Programming

Bad Cookies?

26

- Urban myth: tracking, violate privacy
- Reality:
 - ▣ cookies are relatively harmless
 - ▣ can only store a small amount of information

CSC443: Web Programming

Create a cookie

27

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

PHP

```
<?php  
setcookie("user", "Harry Potter", time()+3600);  
?>
```

```
<html>  
.....
```

PHP

CSC443: Web Programming

Retrieve a Cookie Value

28

```
<?php  
// Print a cookie  
echo $_COOKIE["user"];  
  
// A way to view all cookies  
print_r($_COOKIE);  
?>
```

PHP

CSC443: Web Programming

Delete a Cookie

29

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

PHP

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

CSC443: Web Programming

PHP sessions

31

- Web servers don't keep track of information entered on a page when the client's browser opens another page
 - ▣ Difficult to do anything involving the same information across several pages
- Sessions help by maintaining data during a user's visit by storing data that can be accessed from page to page in a site.
- One can use session variables for storing information

CSC443: Web Programming

Sessions

30

- What is a session?
 - ▣ a combination of a server-side cookie and a client-side cookie,
 - ▣ the client-side cookie contains only a reference to the correct data on the server.
- when the user visits the site:
 - ▣ their browser sends the reference code to the server
 - ▣ the server loads the corresponding data.

CSC443: Web Programming

PHP sessions

32

- Servers keep track of users' sessions using a *session identifier*
 - ▣ Generated by the server when a session starts and is then used by the browser when it requests a page from the server.
 - ▣ The session ID can be sent through a cookie (the default behavior) or by passing the session ID in the URL string.
- Sessions only store information temporarily
 - ▣ If one needs to preserve information, say, between visits to the same site, there is a need to consider methods such as using a cookie or a database

CSC443: Web Programming

Start/end a session

33

```
bool session_start ( void )
bool session_destroy ( void )
```

PHP

- All your session data is stored in the session superglobal array, `$_SESSION`

```
$_SESSION['var'] = $val;
$_SESSION['FirstName'] = "Jim";
```

PHP

CSC443: Web Programming

Session Tracking

34

- A session is the time span during which a browser interacts with a particular server
- For session tracking, PHP creates and maintains a session *tracking id or variable*
 - Create the id with a call to `session_start` with no parameters
 - Subsequent calls to `session_start` retrieves any session variables that were previously registered in the session
- Session variables are set the PHP global variable: `$_SESSION`

CSC443: Web Programming

Storing a Session Variable

35

- Store and retrieve session variables using 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>
```

CSC443: Web Programming

PHP sessions: Example

36

```
<?php
session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Session example</title>
</head>
<body>
<?php
include_once ('object.php'); // Includes definition of the Person class
$_SESSION['hello'] = 'Hello world';
echo $_SESSION['hello'] . "<br/><br/>\n";
$_SESSION['one'] = 'one';
$_SESSION['two'] = 'two';
$me = new Person("Russ", 36, 2892700);
$_SESSION['name'] = $me->get_name();
echo "Testing " . $_SESSION['one'] . ", " . $_SESSION['two'] . ", " . $me->get_number() . ".\n";
?>
</body></html>
```

[view the output page](#)

CSC443: Web Programming

Using session variables

37

- Once a session variable has been defined, you can access it from other pages.

```
<?php
session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
<title>Session example 2</title>
</head>
<body>
<?php
echo "Welcome to a new page ". $_SESSION['name'] ."!<br/>\n";
echo "Hope you enjoy your stay! <br/>";
?>
<p>Back to regular HTML text...
</p>

</body>
</html>
```

CSC443: Web Programming

[view the output page](#)

More on session variables

39

- Need to include a call to the `session_start()` function for each page on which you want to access the session variables.
- A session will end once you quit the browser (unless you've set appropriate cookies that will persist), or you can call the `session_destroy()` function.
 - Note, however, even after calling the `session_destroy()` function, session variables are still available to the rest of the currently executing PHP page.
- The function `session_unset()` removes all session variables.
 - If you want to remove one variable, use the `unset($var)` function call.
- The default timeout for session files is 24 minutes. It's possible to change this timeout.

CSC443: Web Programming

Session Tracking

38

- To create a session variable, use `session_register`
- The only parameter is a string literal of the name of the session variable (without the dollar sign)
 - Example: count number of pages visited
 - Put the following code in all documents

```
session_start();
if (!isset($page_number))
    $page_number = 1;
print("You have now visited $page_number");
print(" pages <br />");
$page_number++;
$session_register("page_number");
```

CSC443: Web Programming

Deleting all session variables

40

```
<?php
session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
<title>Session example 3</title>
</head>
<body>
<?php
echo "Deleting all session variables using session_unset(); <br/>\n";
session_unset();
echo "Now the session variables are gone. <br/>\n";
if (isset($_SESSION['name']))
    { echo $_SESSION['name'] . " <br/>\n"; }
else
    { echo "Session variable is not here."; }
?>

</body>
</html>
```

[view the output page](#)

CSC443: Web Programming

Cookies vs Sessions

41

- Cookies can be set to a long lifespan
- Sessions are stored on the server, i.e. clients do not have access to the information you store about
- Session data does not need to be transmitted with each page; clients just need to send an ID and the data is loaded from the local file.
- Sessions can be any size you want because they are held on your server,

CSC443: Web Programming

More on PHP and SQL

43

- To increase security of your PHP/SQL setup (and to make it easier to change the database you use), it's recommended that you build an "include" file that will have the information you use to connect to the database.

```
<?php
/* Save this as db_login.php (or whatever you like) and
   include it in your php script. */

// Here's the information to connect to the database.
$db_host = 'mysql';
$db_database='martin';
$db_username='martin';
$db_password='';
?>
```

- If someone tries to view this file through their browser, the PHP interpreter will process it and return a blank page to the user (there's no HTML in the file).

CSC443: Web Programming

42

Quick Notes on PHP and MySQL

CSC443: Web Programming

Connecting to the database

44

- Now you can build your PHP script as follows (using the commands that we discussed previously):

```
<?php
include_once ('db_login.php');
$connection = mysql_connect($db_host, $db_username, $db_password);
if (!$connection)
{
    exit("Could not connect to the database: <br/>" .
        htmlspecialchars(mysql_error()) );
}
else {
    // more statements here. . .
}
?>
```

- Note: The function 'htmlspecialchars()' converts special characters in a string into their HTML escape sequences (like '&' into '&') and so forth). This can also be used to increase the security of your code by and help thwart attacks on your database by passing it information that your client has submitted before trying to insert it in your database.

CSC443: Web Programming

MySQL queries inside of PHP

45

- Your MySQL queries from a PHP script are the same as they are as when you're using the MySQL program from the command line with one difference... the queries do not have a semi-colon at the end.
- Aside from this difference, all of the regular SQL commands and keywords are available when you perform your queries.
- You can create new tables, alter, and delete them from inside of a PHP script, and you can also insert and delete rows of tables as normal too. For any such operation, you'll likely want to check to see if it's successful (especially if you're trying to extract data from the database).

```
<?php
// Assuming a valid database connection has been established.
// Build the query string by assigning variables...
$query = $select . $column . $from . $tables . $where;
$result = mysql_query($query);
if(!$result) {
    exit("Could not query the database: <br/>" .
        htmlspecialchars(mysql_error()) );
}
else {
    // process the data
}
?>
```

CSC443: Web Programming

Processing the results of a query

46

- There are two main PHP methods to fetch the results of an SQL query, these being 'mysql_fetch_row()' and 'mysql_fetch_array()'.

```
<?php
// Assuming a database connection, and a valid query
string.
$result = mysql_query( $query );
while ($result_row = mysql_fetch_sql($result)) {
    echo $result_row[2] . '<br/>';
}
?>
```

- The 'mysql_fetch_row()' command fetches the query results as an enumerated array, one row at a time, returning FALSE when there are no more rows (ending the 'while' loop in this case).

CSC443: Web Programming

Processing the results of a query (cont.)

47

- The other command can get a row of results as an associative array. It takes a result as its first parameter, and an optional second parameter as a way to bind the results in the array.
- If MYSQL_ASSOC is specified, the results are indexed using the column names in the query. If MYSQL_NUM is specified, then the number starting at zero accesses the results. The default value MYSQL_BOTH returns an array with both types.

```
while ( $row = mysql_fetch_array($result, MYSQL_ASSOC) ) {
    echo $row["title"] . '<br/>';
}
```

- Using a statement like 'mysql_fetch_array(\$result, MYSQL_NUM)' is essentially equivalent to the statement 'mysql_fetch_row(\$result)' as they both return arrays stored with numerical indices.
- The 'mysql_fetch_array()' command can be used to save memory by specifying, say, MYSQL_ASSOC instead of the default value.

CSC443: Web Programming