

# PHP FILES

Input and Output

# Announcements

2

- ❑ **Do not use sudo on a UR computer**
  - ❑ This includes Betaweb
  - ❑ You do not need to install Git on Betaweb—it is already there.
  - ❑ If you do need to install something; I will need to ask the CS lab staff to do it.

# New Scrum Masters

3

Backslash

ELIZABETH

BRADLEY

C.O.D.E.

MATTHEW

GRAICHEN

Cellar

EVAN

BASTA

ContraWeb

YI

YU

Hacklemore

CHRISTOPHER

DAWSON

Lanister

JAMIE

ALEXANDER

Llama

CONNOR

DENT

Scrum City

CHRISTOPHER

BELL

Sk3m Team

SARAH

HARARI

SqlThePrql

CIARAN

DOWNEY

Synapps

WOLF

HONORE

Tautology

THOMAS

CRAW

Team RNG

JINZE

AN

# Coding Standards

4

- Code to be read by humans
  1. Indent your code in a reasonable fashion
  2. Use meaningful names
  3. Write your code so it is easy to read—i.e. it does what it looks like it does.
  4. Avoid redundancy.
  5. Add comments to clarify tricky bits.
    1. Add a comment for each page saying what the page is.
    2. Do not write unnecessary comments—eventually every comment is a lie.

# Coding Standards

5

- But it needs to work: validate with XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html;  
charset=us-ascii" />
```

...

- Always check that your code is error free.

# Contents of master branch

6

- On February 24, the TAs will start grading the assignments based on your master branch
  - ▣ They will check out the master branch and load it onto their AWS server.
  - ▣ They will then check the code from there.
- Make sure that everything your assignment needs is in the master branch. (i.e. don't use relative branches that will break on another server.)
- Note extra credit on Splashscreen if you want it.

# Quiz

7

```
$ pwd
/home/martin/test
$ ls
file1.txt file2.txt file3.txt
$cat file 1
Hello
World
$
```

1. What is the value of `$a` after `$a = glob(file*)`;
2. What is the value of `$b` after `$b = file($a[0])`;
3. What is the value of `$b` after `$b = get_file_contents($a[0])`;
4. Would you use a session or a cookie for data you don't want the user to see?
5. What do you see `include(no_file)? Require(no_file)`

# Quiz

8

1. What is the value of `$a` after `$a = glob(file*);`
  - ▣ `$a[0] = file1.txt, $a[1] = file2.txt. $a[2] = file3.txt`
2. What is the value of `$b` after `$b = file($a[0]);`
  - ▣ `$b[0] = "hello" $b[1] = "world"`
3. What is the value of `$b` after `$b = get_file_contents($a[0]);`
  - ▣ `$b = "hello/nworld"`
4. What is the difference between a cookie and a session?
  - ▣ A cookie lives on the client computer until deleted; a session lives on both the client and server as long as the connection lasts.
5. What do you see if you have `include("no_file")`? `Require("no_file")`
  - ▣ Include: normal page; require: nothing

9

# PHP include file

# PHP Include File

10

- 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

# include () example

11

```
<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>
```

CSC 210

PHP

12

# PHP File Input/Output

# PHP file I/O functions

13

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

# Reading/writing files

14

<b>contents of foo.txt</b>	<b>file("foo.txt")</b>	<b>file_get_contents("foo.txt")</b>
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

# Reading/writing an entire file

15

```
# 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

# Appending to a file

16

```
# 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!~

# The `file` function

17

```
# 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*

# Unpacking an array: `list`

18

```
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

# Fixed-length files, `file` and `list`

19

```
Xenia Mountrouidou  
(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

# Splitting/joining strings

20

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

*PHP*

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

*PHP*

- `explode` and `implode` convert between strings and arrays

# Example explode

21

```
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*

# Reading directories

22

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

# Example for glob

23

```
# 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

# Example for glob

24

```
# 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

# Example for scandir

25

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

*PHP*

- .
- ..
- 2009\_w2.pdf
- 2007\_1099.doc

*output*

26

# PHP Exceptions

# Exceptions

27

- 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

# Exception example

28

```
<?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*

# Exception example (cont.)

29

```
<?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();
}
CSC 210
?>
```

PHP

# PHP larger example

30

- 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.

CSC 210

**Evan Esar**

31

# PHP cookies and sessions

# Cookies

32

- 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.



# Bad Cookies?

33

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

# Sessions

34

- 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.

# Cookies vs Sessions

35

- ❑ Cookies can be set to a long lifespan
- ❑ Cookies work smoothly when you have a cluster of web servers
- ❑ 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,

# Create a cookie

36

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

*PHP*

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

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

*PHP*

# Retrieve a Cookie Value

37

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

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

*PHP*

# Delete a Cookie

38

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

*PHP*

# Start/end a session

39

```
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*