



## CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

### DIGH 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2014

Week 5

## PHP and MySQL

### Basic HTML output & queries - Part 3

- output all users currently registered in the framework
- then, output content per current user

[PHP Example - Less Redundancy](#) | [PHP Code Example](#)

**\*\*TO DO\*\* - ABSTRACT FUNCTIONS & ADD INCLUDES TO COMBINE THE FOLLOWING**

- output all content
- output all users

## PHP and MySQL

### Code Abstraction - PHP Include()

- allows a script to use functions, code, output etc from another PHP file

```
include('includes/mysql_tools.inc.php');
```

- procedural logic still applies to PHP file
- include() places the code from another PHP file into that position within the current file
- include() allows us to include functions, scripts etc
  - anything that we need to abstract...

## PHP and MySQL

ABSTRACTED FUNCTIONS & ADD INCLUDES TO COMBINE THE FOLLOWING

- output all content
- output all users

[PHP Example - Abstracted](#) | [PHP Code Example](#) ('basicInclude' directory in digh402 GitHub repository)

**\*\*TO DO\*\*** - Basic Error Checking

- check for empty results
- handle errors gracefully

## PHP and MySQL

### Error Handling - Basic ([PHP Manual - Error Handling](#))

- What is error handling in programming?
  - reporting to help developers...
  - error feedback for users...
- How can we handle errors in PHP?
  - die() function or exit() function - [PHP Manual](#)
  - stops script running at point of error
  - custom error handler (see next slide)
- Graceful errors?
  - eg: check user input and logically respond to error
  - trigger\_error("Age must be greater than 21!") - [PHP Manual](#)
  - exceptions (throw, try, catch) - [PHP Manual](#)
- Recording errors
  - saved to error\_log on server
  - custom error\_log() to send email with error number, error string

## PHP and MySQL

### Error Handling - Custom Error Handler

```
<?php
//error handler function
function customError($errno, $errstr)
{
    echo "<b>Error:</b> [$errno] - $errstr";
}

//set error handler
set_error_handler("customError");

//trigger error
echo($test);
?>
```

//sample output for above example  
**Error:** [8] - Undefined variable: test

Example from: [W3 Schools Overview](#)

## PHP and MySQL

### **\*\*TO DO\*\*** - Basic Error Checking

- check for empty results
- handle errors and return feedback for the user

### Handle user error reporting

- basic empty link errors
- empty result set or single empty result
- empty or invalid data returned per DB table row

...

## PHP and MySQL

Basic Error Checking - ~ 10 minutes

- work your way through the site and identify potential points of error in the logic and flow
- go through the code and identify the above points where the code needs to be amended

<http://students.ctsdh.luc.edu/teaching/demos/mysql/basicInclude/>



## PHP and MySQL

### Basic Error Checking - BasicInclude2

- results.php - handle empty 'req' from URL - [Example](#)
- mysql\_connect.inc.php - handle empty 'results' dataset returned from DB - [Example](#)
- content\_viewer.php - [Example](#)
- user\_viewer.php - [Example](#)

Why do we not handle errors in the following include files?

- content\_processor.inc.php
- results\_format.inc.php

[PHP Example - Errors](#) | [PHP Code Example](#) ('basicInclude2' directory in digh402 GitHub repository)

## PHP and MySQL

Further abstraction in the current code - BasicInclude3

- root.inc.php
  - per required directory to allow specification of root directories
- default\_includes.inc.php
  - allows us to store all cross-framework links to include files in one single file
  - eg: MySQL connection & query,
- config.inc.php
  - define 'assets' directory for css & javascript files
  - define 'media' directory for images, video, audio...
  - define MySQL DB settings, tables...

## PHP and MySQL

Further abstraction in the current code

- root.inc.php

- allows abstracted specification of root directories
- can be project root directory or per required directory or often both!
- mainly used to prevent unwanted repetition of directory location in 'include' statements
- eg: location of default includes directory, modules...

Code Example

```
<?php
$root_base = 'modules/base/';
$root_content = 'modules/content/';
$root_user = 'modules/users/';
$root_images = 'media/images/';
$root_includes = 'includes/';
?>
```

## PHP and MySQL

Further abstraction in the current code

- default\_includes.inc.php
  - allows us to store all cross-framework links to include files in one single file
  - eg: MySQL connection & query

Code Example

```
<?php
/*DB config etc*/
include($root_includes.'config.inc.php');
include($root_includes.'mysql_connect.inc.php');
?>
```

## PHP and MySQL

Further abstraction in the current code

- config.inc.php
  - allows us to store all cross-framework links to include files in one single file
  - eg: MySQL connection & query, DB tables, template settings...

Code Example

```
<?php
//database server
define('DB_SERVER', 'localhost');
//database query user login name
define('DB_USER', '402user');
?>
```

- code and files now need to be updated to reflect this latest abstraction of code and settings

## PHP and MySQL

Again, more abstraction - Updated Framework layout and Model

- default framework design components
  - header, sidebar, main content, footer...
- header abstracted to modules/template in framework

[PHP Example](#) | [PHP Code Example](#) ('basicInclude3' directory in digh402 GitHub repository)

...and more to abstract and update.