INLS 672
Web Development 2

PHP Error Handling

UNC
SCHOOL OF INFORMATION
AND LIBRARY SCIENCE

Joan Boone
jpboone@email.unc.edu

Types of errors...

<u>PHP errors</u> (or any programming language)

- Syntax
- Runtime
- Logic

- How to fix? Use built-in or custom error handlers to identify and isolate the error

<u>User “errors”</u>

- Missing required fields
- Wrong data type, size, format
- Value out of range

- How to fix? Write code to validate the data and notify the user of the problem

PHP Errors

XAMPP is a <u>development</u> server

- Not a production server (for this class)
- By default, errors are displayed so you can test, debug, and fix problems in your code
- Displayed errors are also logged to a file located at

  \xampp\apache\logs\error.log

PHP Error Reporting Levels

Configuration information, including error reporting level, is located in **\xampp\php\php.ini**

```
// Report all PHP errors - this is the default for development
error_reporting(E_ALL);

// Report simple running errors
error_reporting(E_ERROR | E_WARNING | E_PARSE);


// Reporting E_NOTICE can be good, too (to report uninitialized
// variables, or catch variable name misspellings ...)
error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);

// Turn off all error reporting - NOT a good idea!
error_reporting(0);
```

PHP Manual: error_reporting

PHP Error Handling

Default error handlers send messages to the browser with a descriptive message, file name, and line number.

Some common errors:

PHP Parse error: syntax error, unexpected end of file

PHP Parse error: syntax error, unexpected '<'

PHP Notice: Undefined offset: 4

PHP Notice: Undefined variable: $mesage

PHP Warning: Division by zero

PHP Warning: Missing argument 1 for perform_action(), called in...

PHP Warning: Wrong parameter count for number_format()

PHP Warning: number_format() expects parameter 1 to be double, string given

PHP Fatal error: Call to undefined function *function-name()*

PHP Fatal error: Maximum execution time of 120 seconds exceeded in *program*

PHP Error Handling

- Users of your web application should never see these error messages
- Through testing and debugging, you fix the code problems that produce the errors
- Some error conditions cannot be anticipated, and require code to intercept the problem
  - PHP Exception handling (try/catch)
  - Create a custom error handler
  - Use basic error handling with the exit() function which displays a message and exits the script

w3schools: PHP Error Handling