Starting To Write PHP Code

April 22nd 2014

Thomas Beebe

Advanced DataTools Corp

(tom@advancedatatools.com)

Advanced DataTools

Tom Beebe

Tom is a Senior Database Consultant and has been with Advanced DataTools for over 10 years. He has been working with Informix since college and is currently the lead consultant for Networking, Unix System Administration and Web Development. Tom is Project Manager and lead developer on a variety of Web Development projects.

Contact Info:
tom@advancedatatools.com
www.advancedatatools.com
703-256-0267 x 106

About This Webcast Series

Four Webcasts:

- March 18th – Getting started
- April 22nd – Data structures, objects, functions and database access.
- May 20th – Introduction to web focused programming. Creating simple and advanced forms
- June 17th – Putting it all together, assorted advanced topics.
- Make sure to register for them

Getting Going Part 1

- In this session we will cover:
  - Writing PHP as command line scripts
  - Managing data types
  - Classes
  - File Access
  - Functions
  - Covering common functions
  - Common language aspects

Variables

- All variables start with $
- Must start with a letter or underscore but can contain numbers
- Names are case sensitive
- Variables types in PHP do not have to be defined, it is good practice to do so.
- They do not have to be given specific casts.
- They will be automatically defined on first use, the system will determine the most appropriate data type for them depending on how they are being used.
- Once a variable has been used it does stay as that data type unless recast or reused.

Variables - Examples

- $a = 'a';
- $b = 'apples';
- $c = 5;
- $c = $c * 5;
- $d = $c . " " . $b;
  - **print $d; // 5 apples**
- $c+= 5;
- $a++;
  - **print "The value of A is: " . $a; // The value of A is: b**

Variable Scope

- Be default variables are treated as local.
  - **They will not be directly accessible by other functions.**
  - **Are not inherited by child functions.**
- The global keyword employed inside a function will grant access to the global variables named.
  - **function some_function(){**

    global $x, $y, $z;

    }
  - **Full access to modify contents of the variables not just a local copy.**
  - **Calling functions need to have access to the variable to allow children to reference it.**

Built In Variables

- PHP has several built in variables:
  - **$_SERVER – Information about the request made, the requesting client, and the server**
  - **$_SESSION – web based session variables**
  - **$_GLOBAL – an array of global variables.**
  - **$_GET/$_POST/$_REQUEST/$_FILES – Used for web based forms**
  - **$_ENV – System environment variables**
  - **$_COOKIE – Web based cookies**
  - **$argc – Number of arguments passed (CLI)**
  - **$argv – Array of arguments passed**
    - $argv[0] is always the script name

Comments – Use Them

- // Comment 1
- # Comment 2
- /*

  Comment 3

*/

Comparisons

- ==
  - **Close equals**
    - Null
    - False
    - ''
    - 0
- ===
  - **Literal equals**
- !=
  - **Not Equals**

Structure Of A Simple Script

```
<?php
$x = 3;
$x = $x * 5;
print “The value of X is: “. $x .”\n”;
?>
```

What A PHP Script Does On Start

- Scans for syntax errors
- Runs through the file to load in any classes or functions
- Starts from the top executing going down
- Any included files are scanned as soon as the parser hits the required lines.
  - **Syntax errors on included files are checked then.**
  - **Functions and classes are loaded in at that point**

Functions

- Over 1000 built in functions
- Can create your own
- Must start with a letter or underscore but can contains numbers
- Case sensitive
- Cannot be overloaded
- Parameters and return values do not need to have data types defined

Sample Function

```
<?
$x = 3;
$x = add5($x);
print “X is now: “. $x;

function add5($var){
    $var += 5;
 return($var);
 }

 ?>
```

Function Arguments

- Can have as many arguments as you want in a function.
- Can define a default which will be used if that argument is not included in the function call.
- By default arguments pass the value of the variable at time of calling.
- If you prepend the variable with an '&' it passes the reference to the variable so if you modify the value it will be updated in the full scope of that variable.

Function Arguments

```
function foo($var, $x=2){
    return("The value of $var is $x");
}
print foo('Apple', 5');
    The value of Apple is 5
print foo('Apple');
    The value of Apple is 2
print foo();
    PHP Warning:  Missing argument 1 for foo()
```

Arrays

- Arrays can be several types
  - **Numeric indexed $x[0] = 5; $x[1] = 3;**
  - **Associative $x['fruit'] = 'Pear';**
  - **Multidimensional $x[0][1] = 3;**
  - **Mixed Multidimensional $x[3]['apple'] = 5;**
  - **As deep as you need (if you can keep track of it)**
    - $x[3]['apple'][2]['description'] = “Test”;
- Defined just by setting them as arrays.
- New array manually just set it to array();

Arrays

Can even have fully mixed indexes

```
$x[1] = 'apple';
$x['b'] = 'pear';
print_r($x);

Array
(
    [1] => apple
    [b] => pear
)
```

Arrays

- Just remember, just because you can do something, doesn't mean you always should.
- Break up your data into manageable data structures, don't just make a massive array.
- Think of future developers who might have to support your code.

Classes

- Introduced in PHP5
- Contains:
  - **Constants**
  - **Functions (methods)**
  - **Variables (properties)**
- Extremely useful for data management.
- Very helpful writing maintainable code.

Setting Up A Class

- What functions do we need to call
- Handle everything consistently
- Think of it as building an API
- How should the data be handled
- Properly trap and display errors

Public Vs Private

- Public – Direct call
- Private – Can only be called from the class
- Protected – Can only be called from the class but can be modified or called from a class extension
- Can be applied to variables under $this as well as to functions
- Use it to maintain code consistency

Why use a class?

- Standard input and output of system calls
- Make sure there is only a single point for system calls
- Proper trapping of data every time
- Separate the database logic from the code
- Usable across multiple projects

This

- This variable
- Only usable in classes, will throw an error if referenced elsewhere
- Will be global across the class
- It is an object that will map to whatever variable you assign to that object type in code
- $this->var[$x][3] = 'test';
- $this->$var = 'test';

Empty Class

- $info = new stdClass();
- Quick way to have an empty class for use in your code without having a full class declaration

Extensions

- Classes can be extended
- This either adds new functions/abilities to an existing class or allows for overloading of existing functions.
- Can get complicated quickly, but very useful if working on an existing product or working as a team with a standard API or class system.

Special Functions

- __construct – act on creation
- __destruct – act on removal of class item
- spl_autoload_register – Looks in the include directory and loads a class by the defined name

__construct Function

- A function named __construct will be called with the arguments passed to the new class line
- class user {
  - **function __construct($user_id) {**
    - do_something();
  - **}**
- }
- $user_account = new user($user_key);

Loops

- for ($x=0; $x<5; $x++){ }
- while ($x < 5) { $x++; }
- foreach ($info as $key => $var){ }

Error Handling

- Try/catch added in PHP5
- Try { code(); } catch (Exception $e) { fail_function(); }
- Will execute the statement in the try code block.
- If it errors or throws an exception rather then crashing it will set $e to the exception information and execute the code in the catch block.
- Great to work with beginTransaction, commit, rollback.

Some Common Functions

- fopen, fread, fgets, fwrite
- substr, strlen, trim
- explode, preg_split
- str_replace, preg_replace, preg_match
- print_r
- mail()
- die

Links

- http://www.php.net
- http://framework.zend.com/
- http://www.zend.com/
- http://www.iiug.org/opensource
- http://pecl.php.net/package/PDO_INFORMIX
- http://pecl.php.net/package/PDO_IBM
- http://www.openadmintool.org

IIUG 2014

Next week!

Miami Flordia.

Informix Support and Training from the Informix Champions!

- **Informix Consulting Services**
- **Informix Support Services**
- **Informix 24x7 Remote DBA Support**
- **Informix Advanced Training**
- **Informix Performance Tune-up**

Free Informix Performance Tuning Webcasts:

http://advancedatatools.com/Informix/Webcasts.html

Fastest Informix DBA Contest 2014

- **Running July to September 2014**
- **The focus this year is on SQL Performance Tuning**
- **Webcast to kick-off the contest on June 24th, 2014**
- **For more information visit:**
  - **http://www.advancedatatools.com**

Starting To Write PHP Code

April 22nd 2014

Thomas Beebe

Advanced DataTools Corp

(tom@advancedatatools.com)

Advanced DataTools