Function Junction

Anne Rue
Regional Account Manager
Anne.rue@zend.com
(310) 480 4161

Mike Pavlak
Solutions Consultant
Mike.p@zend.com
(815) 722 3454
Agenda

- What are functions in PHP?
- Base functions in PHP
- Home grown function
- Data access function
- Q&A
What are functions in PHP?

- What’s so great about functions?
- Remember life before subroutines?
- Need to organize code
- Before OOP, functions was all they had!
- Also the basis for language
- PHP supports more than 5000 functions
- And you can create them!
- www.php.net/functions
http://php.net/functions

Functions

Table of Contents

- User-defined functions
- Function arguments
- Returning values
- variable functions
- Internal (built-in) functions
- Anonymous functions
Functions come from 1 of 3 places

- Built-in
  - Part of base PHP
- Extensions
  - Components of modules like image functions in GD
- User defined
  - You will create these!

You may have been using functions…

- db2_connect()
- print_r
- strtoupper
- Etc.
Function fundamentals

- Can accept parameters
- Will always return a value
- Maintain scope of control within

- Echo is a construct, not a function
  - Does not return a value
Base PHP Functions
What do functions look like?

**Function call has three major parts:**
- Function Name - Required
- Parameters - Optional
- Return Value – Always returned, default null

```php
$conn = db2_connect('localhost', $UserID, $Pwd);
```
Functions in action - Simple

```php
$UserID = 'MPavlak';
$Pwd = '******';

echo $UserID . "<br><br>";
$UserID = strtoupper($UserID);
 echo $UserID;
```

The output of the code is:

```
MPavlak
MPAVLAK
```
How do you get a date?

```
$currentTime = time(); print_r($currentTime); echo "<br>";
$currentTime = date(r); print_r($currentTime); echo "<br>";
$currentTime = date("m/d/y g:i:sa"); print_r($currentTime); echo "<br>";
echo "Today is $currentTime";
```
More built in functions…

- About 5,000 in base and extensions
- Gettype – Can return the variable type but…
  - Can give you incorrect info like String 0 returns Bool
- Use is_* functions

<table>
<thead>
<tr>
<th>is_array()</th>
<th>is_object()</th>
</tr>
</thead>
<tbody>
<tr>
<td>is_bool()</td>
<td>is_resource()</td>
</tr>
<tr>
<td>is_int()</td>
<td>is_scalar()</td>
</tr>
<tr>
<td>is_numeric()</td>
<td>is_string()</td>
</tr>
</tbody>
</table>
Gettype...

```php
// Scalar
$Int1 = 25;   $str1 = 'The quick brown fox';
$flt1 = 3.141; $bool1=false;

// Complex
$arr1 = array('Dog','Cat'); $obj1 = new stdClass();
$res1 = i5_connect("", "PHPUSER","phpuser"); $null1 = null;

echo "<h3>Scalar variables:</h3>";
echo "Value $Int1 is " . gettype($Int1);
echo "<br>Value $str1 is " . gettype($str1);
echo "<br>Value $flt1 is " . gettype($flt1);
echo "<br>Value $bool1 is " . gettype($bool1);

echo "<h3>Complex variables:</h3>";
echo "Variable value $arr1 is " . gettype($arr1);
echo 'Value $obj1 is ' . gettype($obj1);
echo 'Value $res1 is ' . gettype($res1);
echo 'Value $null1 is ' . gettype($null1);
```
User Defined Functions
User Defined Functions

- PHP version of sub-procedure (subroutine)
- Programmer defines action, input and results
- This is how applications are built
- Modular components, pulled together
Rules for creating functions

- **See php.net for more details**
  - Must have name
  - Parameters are optional, you test inside function
  - Function always returns a value, even if you don’t
  - Function variables have scope
  - Builds suspense for OOP, but not yet…
Let’s create a function

- **Create a pricing function**
  - Called with a price & quantity
  - Will add standard discount based on quantity
    - Qty 1-10 gives 10%
    - Qty 11-50 gives 15%
    - Qty > 50 gives 20%
  - Return adjusted price
function customer_price( $price, $quantity ) {
    switch ( $quantity ) {
        case ( $quantity > 50 ) :
            $discount = .2;
            break;
        case ( $quantity > 10 ) :
            $discount = .15;
            break;
        default:
            $discount = .1;
    }
    $newPrice = $price * ( 1 - $discount );
    return $newPrice;
}
Test the function

```php
$p1 = 5.00; $q1=15; $output=customer_price($p1,$q1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
$p1 = 0.00; $q1=15; $output=customer_price($p1,$q1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
$p1 = 12.00; $q1=9; $output=customer_price($p1,$q1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
$p1 = 10.00; $q1=800; $output=customer_price($p1,$q1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
```

- Retail is 5 and quantity is 15 with discount price at 4.25
- Retail is 0 and quantity is 15 with discount price at 0
- Retail is 12 and quantity is 9 with discount price at 10.8
- Retail is 10 and quantity is 800 with discount price at 8
What if I do no pass a quantity?

- Ask the BA…
- BA states that you should default to retail
- OK, let’s go change the code!
function customer_price( $price, $quantity="zero") {
    switch ($quantity) {
        case ($quantity == "zero"):  
            $discount = 0;
        break;
        case ($quantity > 50):
            $discount = .2;
        break;
        case ($quantity > 10):
            $discount = .15;
        break;
        default:
            $discount = .1;
    }
    $newPrice = $price * (1-$discount);  
    return $newPrice;
}
Test the mod...

```php
$p1 = 5.00; $q1=15; $output=customer_price($p1,$q1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
$p1 = 0.00; $q1=15; $output=customer_price($p1,$q1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
$p1 = 12.00; $q1=9; $output=customer_price($p1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
$p1 = 10.00; $q1=800; $output=customer_price($p1,$q1);
echo "Retail is $p1 and quantity is $q1 with discount price at $output<br>";
```
• Variables have scope, much like RPG ILE
  • Function keeps its own set of variables
  • Global keyword can change all that (RPG III)
  • Better to pass, globals can’t be trusted

• Static variables
  • Persistence from call to call
Parameters

- **Default**
  - Assign value in interface

- **Pass by value**
  - This is the default behavior

- **Pass by reference**
  - This is done with & before the variable in the interface.

- **Variable number**
  - `func_get_args()`
  - `func_num_args()`
  - `func_get_arg(argument_#)`
Return

- Can only return one value
- If you want to return more than one, try array!

- Let’s look at an example that shows all of these...
function example ($str1, & $str2) {
    $a = 0; static $b;
    $a++; $b++;
    $str2 = $str1;
    $c = array($a, $b);
    return $c;
}

$stringin1 = 'Hello Mike'; $stringin2 = 'Hello Scott';
$result = example($stringin1, $stringin2);
print_r($result);
echo '<br>$stringin1 = ' . $stringin1;
echo '<br>$stringin2 = ' . $stringin2;
$stringin1 = 'Hello Jane'; $stringin2 = 'Hello Laura';
echo '<br>;
$result = example($stringin1, $stringin2);
print_r($result);
Data Access Function
• Build PHP script using functions
• Pass values rather than use global variables
• Analyze function activity with Code Profiler
Raw Data in DB2

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jimmy Buffet</td>
<td>IL</td>
</tr>
<tr>
<td>2</td>
<td>Sherlock Holmes</td>
<td>EN</td>
</tr>
<tr>
<td>3</td>
<td>Gregory House</td>
<td>NJ</td>
</tr>
<tr>
<td>4</td>
<td>Dexter Morgan</td>
<td>FL</td>
</tr>
<tr>
<td>5</td>
<td>Mary Shannon</td>
<td>NM</td>
</tr>
</tbody>
</table>
### Short List of Anti-Heros:

<table>
<thead>
<tr>
<th>Customer Number</th>
<th>Customer Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jimmy Buffet</td>
<td>IL</td>
</tr>
<tr>
<td>2</td>
<td>Sherlock Holmes</td>
<td>EN</td>
</tr>
<tr>
<td>3</td>
<td>Gregory House</td>
<td>NJ</td>
</tr>
<tr>
<td>4</td>
<td>Dexter Morgan</td>
<td>FL</td>
</tr>
<tr>
<td>5</td>
<td>Mary Shannon</td>
<td>NM</td>
</tr>
</tbody>
</table>
Function 1 – Database connect

```php
<?php
function Connect() {
    // Standard DB connection to DB2...
    $conn = "*LOCAL"; $name = ""; $pwd = "";
    $i5link = db2_connect($conn, $name, $pwd);
    if (! $i5link)
        echo 'Connection failed: ' . db2_stmt_error() . ' ' . db2_stmt_errormsg();
    return $i5link;
}
```
Function 2 – Query execute

```php
function GetData($i5link) {
    $sql = "SELECT customer_number, customer_name, customer_state from zenddata.customer";
    $stmt = db2_exec($i5link,$sql)
    or die("Failed query:".$d2_stmt_error().":".$d2_stmt_errormsg());

    return $stmt;
}
```
Function 3 – Read and load data

```php
function listData($stmt) {
    // loop di loop through customer recs...
    while($row=db2_fetch_array($stmt)) {
        //print_r($row); echo "<BR>";
        list( $CUSTOMER_NUMBER, $CUSTOMER_NAME, $CUSTOMER_STATE) = $row;
        //include "somefilethatdoesnotexist.inc";
        echo("<TR><TD> $CUSTOMER_NUMBER </TD> <TD> $CUSTOMER_NAME </TD>
                         <TD> $CUSTOMER_STATE </TD>";
    }
}```
```php
$connection = Connect();
$resultset = GetData($connection);
// Setup table header...
echo '<TABLE BORDER="6" BORDERCOLOR=RED><TBODY BGCOLOR="yellow">';
echo '<TR><TD>Customer Number</TD><TD>Customer Name</TD><TD>State</TD></TR>';
listData($resultset);

echo '</table>';  
db2_close($connection);
?>
</html>
```
If time permits, demo code profile, etc.

• Jump out to demo
New book, new printing, same great stuff!

Kevin Schroeder from Zend’s Global Services Group with Jeff Olen, co-author of…

Get yours at MCPressonline or at fine bookstores everywhere
Thank you!!

Questions?

Anne.rue@zend.com
(310) 480 4161
Mike.pavlak@zend.com