

The Juggling Librarian Teaches PHP



- Lisa Bartle, Reference Librarian
- Cal. State University, San Bernardino
- LBartle@csusb.edu
- 909-537-7552
- www.abc-lit.com
- www.dawcl.com

Presentation given at the California Library Association Conference in Pasadena, CA on Nov. 6, 2005. Sponsored by the IT section of CLA.

Assumptions for this Class

1. You know html code (e.g., tables, forms).
2. You understand the basics of databases (e.g., fields, records, truncation).
3. A little programming experience in any language is useful for this class but not essential. (For example, you know what a variable is.)

Goals for this Class

1. Not to teach everything about PHP, but provide the basic knowledge for an online database.
2. Explain just the code, not setting up a PHP server.
3. Provide a solid grounding so the books are not as formidable.
4. Marketing for the IT section of CLA.
5. Have a good time.

SQL Queries

```
SELECT [field],[field] from [table] WHERE  
[field] [has parameter] [input] ORDER BY  
[field] [ASC or DESC];
```

Database: catalog

Table: books

Fields: title, author, published_year, ISBN

Example searches:

- SELECT title, author, published_year from books where published_year > 2000;
- SELECT title, author, published_year from books where published_year > 2000 and published_year < 2003;
- SELECT title, author, published_year from books where published_year BETWEEN "2000" and "2003";
- SELECT * from books where title IN "Great Expectations";
- SELECT * from books where title LIKE "Great%";
- SELECT * from books where title LIKE "%Great";
- SELECT * from books where title LIKE "%Great%" ORDER BY title ASC, author ASC;

Welcome back
to our regularly scheduled
presentation of PHP ...

Let's Get Down to Business

- PHP can be embedded in html files. The only difference is the file will end in the file extension PHP.
- Though there are alternatives to some of the examples provided, I give the safest code choices that will not require short tags to be turned on at the server level, nor will these examples conflict with XML or XHTML syntax.

PHP Code Tag

CLA PHP presentation

Hello librarians!

- HTML can be outside of PHP coding.
- HTML can be inside of the code, but must be inside the quotation marks.
- PHP statements end in a semi-colon.

```
<h2>CLA PHP  
presentation</h2>  
<?php  
print "<p>Hello  
librarians!</p>";  
?>
```

- PHP code starts <?php
- PHP code ends ?>

Example1.php

Outputting Data

- Echo()-- a language construct or structure for single argument.
- Print()-- a function, which can have multiple arguments.
- Printf()—outputs according to a format (won't cover here)
- Sprintf() —for formatting and displaying numbers as dollars (won't cover here)

```
<?php  
Echo "CLA PHP Presentation";  
Print "<P>Hello librarians."  
?>
```

```
CLA PHP Presentation  
  
Hello librarians.
```

Example2.php

Variables

A variable always begins with a \$ then the variable name. Variables are case sensitive. Do not start a variable with a number.

\$book

\$author

\$amount

\$Amount

\$AMOUNT

Variable Example

CLA PHP Presentation

Hello librarians!

Today's date is Nov. 6, 2006

```
<html><body>
<h2>CLA PHP
  Presentation</h2><br>
<?php print "<p>Hello
  librarians!</p>"; ?><br>
<?php $date="Nov. 6,
  2006";?>
<p>Today's date is
<?php
Print $date;
?>
</body></html>
```

Comments

- `//` beginning each line inside the PHP code
- `#` beginning each line inside the PHP code
- `/*` beginning and `*/` ending several lines inside the PHP code

```
<?php
//Purpose: to create a form for patrons'
questions
?>
```

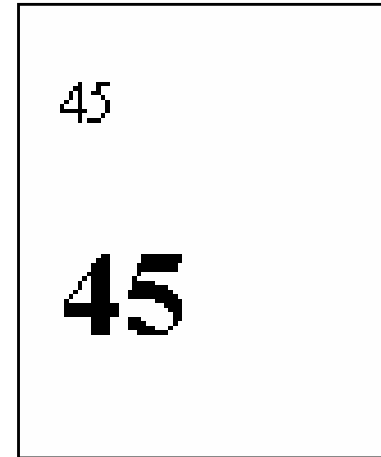
```
<?php
#Purpose: to create a form for patrons'
questions
?>
```

```
<?php
/*
Purpose: to create a form for patrons'
questions
Author: Lisa Bartle
Date: Nov. 6, 2006
*/
?>
```

Comment Examples

Arithmetic Operators

```
<?php
$a=15;
$b=30;
$total=$a+$b;
Print $total;
Print "<p><h1> $total </h1>";
//total is 45
?>
```



45

45

\$a - \$b	//subtraction
\$a * \$b	//multiplication
\$a / \$b	//division
\$a += 5	//\$a = \$a +5 Also works for *= and /=

Example4.php

Concatenation

- Use a period to join strings into one.

```
<?php
```

```
$string1="Hello";
```

```
$string2="Librarians";
```

```
$string3=$string1." ".$string2;
```

```
Print $string3;
```

```
?>
```

Hello Librarians

Escaping the Character

- If the string has a set of double quotation marks that must remain visible, use the \ [backslash] before the quotation marks to ignore and display them.

```
<?php  
$heading="\\"Lisa's Books\\"";  
Print $heading;  
?>;
```

"Lisa's Books"

Example6.php

If ... Then ... Else

If (condition)

{

 Statement;

}

Else

{

 Statement;

}

```
If ($user=="Lisa")
```

```
{
```

```
    Print "Hello Lisa.";
```

```
}
```

```
Else
```

```
{
```

```
    Print "You aren't
```

```
Lisa.";
```

```
}
```

No THEN in PHP

While Loops

While (condition)

{

Statement;

}

```
Hello Lisa. Hello Lisa. Hello Lisa. Hello Lisa.
```

```
$count=0
```

```
While ($count<5)
```

```
{
```

```
Print "Hello Lisa.";
```

```
$count=$count + 1;
```

```
#Another way to say
```

```
#to say that is
```

```
#$count++;
```

```
}
```

Date Display

2005/11/06

```
$datedisplay=date("yyyy/m/d");  
print $datedisplay;  
# If the date is November 6, 2005  
# It would display as 2005/11/06
```

Sunday, November 6, 2005

```
$datedisplay=date("l, F m, Y");  
print $datedisplay;  
# If the date is November 6, 2005  
# It would display as  
# Sunday, November 6, 2005
```

Example8.php & example9.php

Month Format Symbols

M	Jan
F	January
m	01
n	1

Day & Date Format Symbols

Day of Month	d	01
Day of Month	j	1
Day of Week	l	Monday
Day of Week	D	Mon

Include Files

Include “/path/to/filename.php”;

Include “/usr/local/lib/php/includes/connect.php”;

This inserts files; I mean oft repeated code stored in files, into the current code. This will prove useful and protective once you connect to a database, as well as for other repeated functions.

`include (“footer.php”);`

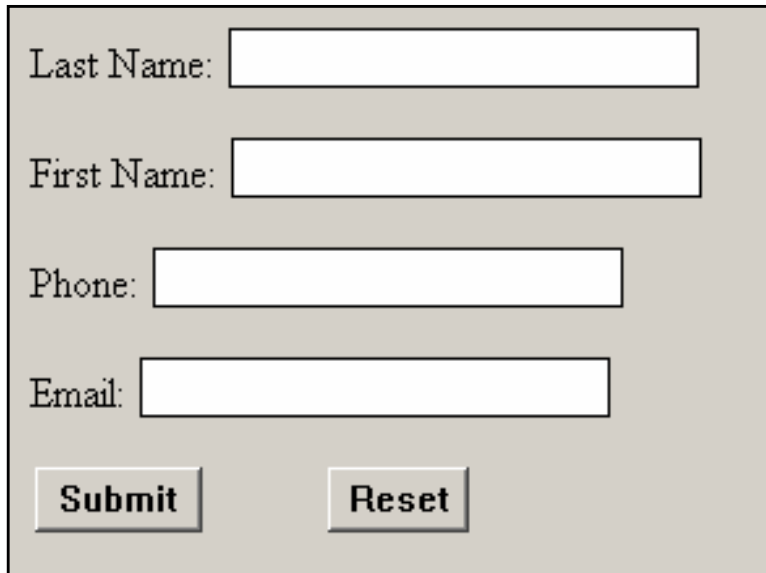
The file footer.php might look like:

```
<hr SIZE=11 NOSHADE WIDTH="100%">  
<i>Copyright © 2001-2006 Lisa R. Bartle</i></font><br>  
<i>ALL RIGHTS RESERVED</i></font><br>  
<i>URL: http://www.abc-lit.com/</i></font><br>
```

Let's look at the files

- Start.html
- Start.php

search.html



Last Name:

First Name:

Phone:

Email:

Database name: addressbook.sql

Table name: people

Field names: lastname,
firstname, phone, email

File name: search.html

```
<html><head></head>
<body>
<p>Last Name: <input
type="text" size="20"
name="firstname"></p>
<p>First Name: <input
type="text" size="20"
name="firstname"></p>
<p>Phone: <input type="text"
size="20" name="phone"></p>
<p>Email: <input type="text"
size="20" name="email"></p>
<p><input type="submit"
name="submit_search" />
<input type="reset" /> </p>
</body></html>
```

Alter Query Form

```
<form action="search.php" method="post">
```

```
<html><head></head>
```

```
<body>
```

```
<form action="search.php" method="post">
```

```
<p>Last Name: <input type="text" size="20" name="firstname"></p>
```

```
<p>First Name: <input type="text" size="20" name="firstname"></p>
```

```
<p>Phone: <input type="text" size="20" name="phone"></p>
```

```
<p>Email: <input type="text" size="20" name="email"></p>
```

```
<p><input type="submit" name="submit_search" />
```

```
<input type="reset" /> </p>
```

```
</body></html>
```

Names and option values are passed off automatically to the PHP.

Keyword word or phrase:

(Implicit phrase search/box)

Type of material:

Publication year:

 through

Sort results?

Submit Query

Reset

Database name: catalog.sql

Table name: materials

Field names: books, author,
format, publication_year

File name: search.html

```

<html><head><title>search</title></head><body>
<form action="search.php" method="post">
<table border="0" width="748">
  <tr><td width="213">Keyword word or phrase:<br />
    <font size="2">(Implicit phrase search/box)</font></td>
    <td width="525"><input type="text" size="19"
      maxlength="25" name="keyword" /></td></tr>
  <tr><td width="213">Type of material:</td>
    <td width="525"><select name="format" size="1">
      <option value="">Doesn't matter</option>
      <option value="book">Book</option>
      <option value="periodical">Periodical</option>
      <option value="microfiche">microfiche</option>
      <option value="vhs">VHS</option>
      <option value="dvd">DVD</option>
    </select></td></tr>
  <tr><td width="213">Publication year:</td>
    <td width="525"><select name="published_low" size="1">
      <option value="2004">2004</option>
      <option value="2005">2005</option>
      <option value="2006">2006</option>
    </select> through <select name="published_high" size="1">
      <option value="2006">2006</option>
      <option value="2005">2005</option>
      <option value="2004">2004</option>
    </select></td></tr>
  <tr><td width="213">Sort results?</td>
    <td width="525"><select name="sort_by" size="1">
      <option value="published">Publication Year</option>
      <option value="periodical">Periodical</option>
      <option value="author">Author</option>
      <option value="title">Title</option>
    </select> </td></tr>
  <input type="submit" name="submit_search" />
  <input type="reset" /></td></tr></table></body></html>

```

search.php

- Creates connection to database
- Alters input for safety
- Creates SQL query
- Displays the results
- Close the connection to database

Assign Variables

```
<?php
# Create connection to database – define variables
$host="localhost";
$username="myusername";
$password="mypassword";
$database="catalog";
$tablename="materials";
#
# But it isn't very safe to have this info. in the
# script so use an include file and put it elsewhere.
```

Create connection to account

Create connection to database

```
$connection=mysql_connect("address",  
"myusername", "mypassword") or die  
("message");
```

```
$result=mysql_select_db($database) or  
die("Database could not be selected");
```

Create Connections

```
include ("start.php");  
#Create connection  
$connection=mysql_connect($host,$uname,$pass) or die  
("Database connection failed.");  
$result=mysql_select_db($database) or die("Database could  
not be selected");
```

Alter Input for Safety

```
# Remove tags from data  
$keyword=trim($keyword);  
$keyword=strip_tags($keyword);
```

```
# Make everything lowercase. MySQL ignores case  
anyway, so this makes parsing easier.  
$keyword=strtolower($keyword);
```

Create Boolean String

Insert an initial plus-sign so the first term or phrase is included as a mandatory.

```
$addplus="+";
```

```
$keyword=$addplus.$keyword;
```

Replace all the AND's with +'s, so that fulltext can understand it.

The spacing is important here.

```
$keyword=str_replace(" and ", " +", $keyword);
```

OR is assumed, so just pull those out.

```
$keyword=str_replace(" or ", " ", $keyword);
```

look for NOTs

```
$keyword=str_replace(" not ", " -", $keyword);
```

Some people may put AND NOT, or OR NOT

Since we already replaced the ANDs and ORs, this will look weird.

```
$keyword=str_replace(" +not ", " -", $keyword);
```

Set Up Alternative Sorting

```
# Change ASC and DESC display based on if published is the sort.  
# If they choose to sort by year, it should display  
# most recent to less so, or DESC.  
# If they choose to sort by author, title, etc., it will  
# display alphabetically, or ASC.
```

```
If ($sort_by=="Published")  
{  
    $order=" DESC";  
}  
else  
{  
    $order=" ASC";  
}
```

Create SQL Query

```
SELECT [field],[field] from [table] WHERE [field] [has  
parameter] [input] ORDER BY [field] [ASC or DESC];
```

```
SELECT * from books where title LIKE "%Great%" ORDER  
BY title ASC;
```

```
$query="SELECT * from books where title LIKE  
\ "%Great%\ " ORDER BY title ASC";
```

Create SQL Query

#Create SQL query

```
$query="SELECT * from " . $tablename. "  
WHERE (author LIKE \"%\".$keyword.\"%\" or  
title LIKE \"%\".$keyword.\"%\") and  
format=\"$format\" and publication_year>=  
\"\".$published_low.\"\" and published_year<=  
\"\".$published_high.\"\" order by " . $sort_by . "  
DESC";
```

Add This to Your Notes

True Boolean query string:

```
$query="SELECT * from " . $tablename . " where  
MATCH(Abstract, Title, Author) AGAINST  
('$keyword' IN BOOLEAN MODE) and  
Periodical like \"%\" . $Periodical. \"%\" and  
Published >= \"\" . $Published_Low. \"\" and  
Published <= \"\" . $Published_High. \"\" order by "  
.$sort_by . $order;
```

\$Result & \$Numrows

```
$result=mysql_query($query) or  
    die("Couldn't execute query.");  
$numrows=mysql_num_rows($result);
```

- \$result is an array where the results are stored.
- \$numrows is the number of rows, or records, in the result of the query.

Display Results to User

```
$counter=0;
while ($row=mysql_fetch_array($result)) # filling-in data from results
{
$counter=$counter+1;
echo $counter. "
".$row["Author"]."&nbsp;\\".$row["Title"]."\\"&nbsp;<i>".$row["Peri
odical"]."</i>&nbsp;".$row["Source"]."<p><blockquote>
<b>Abstract</b>:&nbsp;".$row["Abstract"]."&nbsp;(<i>".$row["Con
tributor"]."</i>)</blockquote><p>&nbsp;<p>";
}
```

Add This To Notes
Close Connection

```
mysql_close($connection);
```

Functions Covered

mysql_connect()

trim()

strtolower()

time()

mysql_query()

mysql_fetch_array()

match() against (,in boolean mode)

mysql_select_db()

strip_tags()

str_replace()

include()

mysql_num_rows()

mysql_close()

Keep Track of Names

- Database name: catalog.sql
- Table name: materials
- Field names: title, author, format, publication_year
- Form names: keyword [title, author], format, published_low [publication year], published high [publication year], sort_by

End of PHP
Beginning of MySQL

MySQL

- A database that is fast, free, reliable RDBMS that is very often found matched with PHP.
- It can run on Unix, Windows, and MacOS.
- www.mysql.com announced 24 October 2005 that 5.0 is available for download.
- ABC-Lit* is run using **MySQL 4.0.13**

MySQL Table Structure

```
-- Table structure for table `Citations`  
CREATE TABLE `Citations` (  
  `Periodical` varchar(100) default NULL,  
  `Source` varchar(55) default NULL,  
  `Title` varchar(255) default NULL,  
  `Author` varchar(255) default NULL,  
  `Abstract` text NOT NULL,  
  `Contributor` varchar(5) NOT NULL default "",  
  `Published` int(11) NOT NULL default '2004',  
  `ID` int(11) NOT NULL auto_increment,  
  UNIQUE KEY `ID` (`ID`),  
  FULLTEXT KEY `Title` (`Title`),  
  FULLTEXT KEY `Author` (`Author`),  
  FULLTEXT KEY `Abstract` (`Abstract`)  
) TYPE=MyISAM AUTO_INCREMENT=2880 ;  
-- Dumping data for table `Citations`
```

MySQL Data

INSERT INTO `Citations` VALUES ('Papers : Explorations into Children's Literature', 'v.6(1), April 1996 : 05-11', '"You have to get used to things as they are": Writing the Child as Victim in Nina Bawden's <I>Squib</I>.', 'Strain, Elizabeth', 'Strain examines the children protagonists in Bawden's <I>Squib</I> (1971) as examples of her attempt "to fuse the demands of a more stringent realism with a desire to protect the sensibilities of her pre-adolescent implied reader." She finds that Bawden does not entirely succeed because "too many of the troubling aspects of damaged childhood are sealed over."', 'lrb', 1996, 1);

INSERT INTO `Citations` VALUES ('Papers : Explorations into Children's Literature', 'v.6(1), April 1996 : 12-19', 'The Writer in the Book: Four Narratives on Writing.', 'Dunscombe, Kordula', 'Dunscombe looks at four novels focused on the act of writing: Gardam's <I>A Long Way from Verona</I> (1971), Cleary's <I>Dear Mr. Henshaw</I> (1983), Fine's <I>Book of the Banshee</I> (1991), and Chambers' <I>Breaktime</I> (1978). She discusses how each attempts "to represent, explore, impart an understanding of, and possibly foster in the reader the act of writing."', 'lrb', 1996, 2);

INSERT INTO `Citations` VALUES ('Papers : Explorations into Children's Literature', 'v.6(1), April 1996 : 20-30', 'Parallel Ideologies: An Exploration of the Ideologies of Childhood and Poetry.', 'Halliday, Alison', 'Halliday uses <I>The Puffin Book of Twentieth-Century Children's Verse</I>, edited by Brian Patten, to compare similar ideologies of childhood and poetry. Each, she says, is an innate part of the human experience; each is elevated and protected; each is marginalized and subordinated in actual practice and study. Patten's anthology, in its structure, organization, selection, and especially in the illustrations, reflect this view of childhood.', 'lrb', 1996, 3);

Other Cool Things About PHP

Open source – It's free!

Works well with Apache – also open source.

Create and manipulate Macromedia Flash, image, and PDF files.

Parse complex strings.

Communicate with many credit card processing.

Communicate with many protocols, including IMAP, POP3, NNTP, and DNS.

Native database support for over 25 db products including dBase, FilePro, FrontBase, IBM DB2, Hyperwave, mSQL, MySQL, Sybase, Unix dbm, and many others.

The History of PHP

1. PHP began in 1995 when Rasmus Lerdorf developed a Perl/CGI script toolset he called the Personal Home Page or PHP.
2. PHP 2 released 1997 (PHP now stands for Hypertext Processor). Lerdorf developed it further, using C instead. His goal was to incorporate code alongside HTML
3. PHP 3 released in 1998 (50,000 users)
4. PHP 4 released in 2000 (3.6 million domains). Considered debut of functional language and including Perl parsing, with other major features.
5. PHP 5.0.0 released July 13, 2004 (113 libraries/>1,000 functions with extensive object-oriented programming)
6. PHP 5.0.5 released Sept. 6, 2005 for maintenance and bug fixes.

Recommended Texts for Learning PHP

Larry Ullman's books from the Visual Quickpro series

PHP & MySQL for Dummies

Beginning PHP 5 and MySQL: From Novice to Professional
by W. Jason Gilmore

(This is more advanced and dense than the others, but great to read once you've finished the easier books. One of the best definition/description of object oriented programming I've read.)

Web sites for more information

- www.php.net (not php.org, which is a commercial site selling php software)
- www.phpfreaks.com

Thank you



- Lisa Bartle, Reference Librarian
- Cal. State University, San Bernardino
- LBartle@csusb.edu
- 909-537-7552
- www.abc-lit.com
- www.dawcl.com