Topic 6: Introduction to Server Side Programming with PHP/MySQL

4 Lecture Hrs, 4 Practical Hrs.

Learning Objectives

By completing this topic, the learner should be able to:

Knowledge and Understanding

1. Discuss the uses of PHP.
2. Describe the basic PHP syntax and structures.
3. Describe how PHP programs are used to create web applications
4. Implement MySQl databases.

Skills

5. Write simple form processing applications with PHP
6. Create database publishing websites using PHP and MySQL

Attitude

7. Acquire confidence in use of PHP and MySQL in web application development.

Key Concepts

Application Architecture	This refers to the structure and interrelationship between the components that constitute the software. Web applications have distinctively different architecture from desktop applications.
Software Lifecycle	These are the stages in the development, maintenance and obsolescence of a piece of software. Different software development models have been developed over the years.
Relational Database management Systems (RDBMS)	These are database management software that use related tables to store data. MySQL is a well-known open source RDBMS.
Open Source	Open source software is software where the source code is available to the user and where the user is permitted to modify the source code and create new modified versions of the software. If may or may not be free in monetary terms although in most cases open source is also free (zero price). All the software tools used in this topic are open source and free.
SQL	SQL is the structured query language developed more than 20 years ago as s means for accessing and managing RDBMS data. Though not a programming language, it provides a means to create and modify databases and to manage user privileges.

Introduction

In Topic 5 and 6 we learned about client side programming with Javascript. In Topic 7 and 8 we shall learn about server side programming for the purposes of creating dynamic content such as publishing live data or generating dynamic web-based charts. The tools selected for this is the open source programming language called PHP and the open source data base administration system called MySQL. Both are very popular tools for web application developers and have capability to create, together with the other tools covered earlier in this course, very powerful web applications. We shall learn about processing files on the server side allowing us to create simple applications that can upload data and manipulate text files.

We examine the parameters that compromise the security and performance of web applications and also look at basic web site administration. However, more detailed study of web administration will be provided in Topic 9.

This topic will provide you with the fundamentals of server-side programming and lay the foundation for further study and skill advancement. Good web programmers are sought out worldwide, therefore, this content provides an opportunity to enter into a rewarding career in website development. It is important to remember that the mastery of any programming language typically takes at least two years of constant practice. Hence, your progression and advancement will depend on your determination to keep learning and practice intensively. Welcome to the world of web-based computing.

PHP Programming Basics
You have already encountered Java and Javascript programming and the basic concepts surrounding the process of programming. PHP is not the only server-side programming language out there. Java, Python, C#, PERL and Ruby are other well-known web programming languages. PHP is attractive due to its wide use on the web and its ease of use. It is also an open source tool meaning that it is freely available for use. PHP is very versatile due to a large base of specialised libraries.

A web application has four essential components shown in Figure 1 below. The browser is the client that provides the user access to the application's interface. The browser interacts with the web server using HTTP as we saw in Topic 1. The web server is configured to communicate with a backend server using a scripting language such as PHP. There is a variety of backend servers in use depending on the type of application. Many applications use a database management system such as MySQL as a backend for storing and managing data. A messaging application may use an email server as a backend.

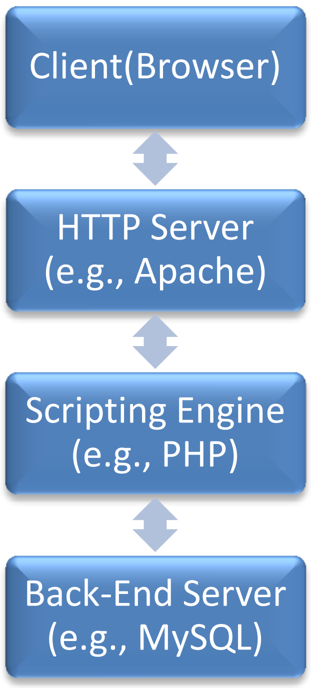

Client(Browser)
HTTP Server (e.g., Apache)
Scripting Engine (e.g., PHP)
Back-End Server (e.g., MySQL)

Figure 1: Architecture of a web application

The web server is configured to recognise programs that should be passed to the scripting engine by the file extension. For example, PHP scripts are saved with the file extension ".php". When the web server is requested to server a PHP file, it passes the file to the scripting engine which is a computer program stored in your computer. The scripting engine processes the PHP code and returns an HTML file as output to the web server that then passes the HTML file to the browser to display**.**

Therefore, the scripting engine acts an intermediary between the web (HTTP) server and the backend server. PHP programs can be self-contained in the sense that no backend server is required. Many of the examples given in this topic do not require a backend server.

A PHP program consists of PHP code sometimes mixed with HTML. The resulting source file has a file extension ".php" and will be saved on the web server. A simple PHP program is illustrated below:

```
<html>
<head>
<title>PHP Example 1 </title>
</head>
<body>
<h1>Simple PHP Program</h1>
<?php
echo "This is a PHP program example";
?>
</body>
</html>
```

Example 1: Simple PHP script

The PHP code is enclosed in the tags "<?php" and "?>". Everything between these two tags is PHP code. Inserting HTML here will cause an error. Outside these tags, you can place HTML. You can have more than one segment of PHP code in your file each enclosed in these tags. Indeed, a PHP program can be broken into more than one part with PHP code in between in the following manner:

```
Html code
<?php
  Php code
?>
Html code
<?php
   More Php code
?>
Html code
<?php
   More Php code
?>
```

WAMP Server

For a PHP application to run, you will need as a minimum to have a web server installed and configured correctly to recognise PHP files. You will also require to have PHP installed. Additionally, if your applications require a backend server, this server will be required as well.

It is possible to install all the above separately for example, using the Apache web server and the MySQL database server but there are easier options. In MS Windows we have all these tools packaged together. The most popular such packages are the WAMP (www.**wampserver**.com/) and XAMP (http://www.apachefriends.org) servers.

The WAMP sever is recommended. After downloading, double-click the downloaded file to install. You need to keep a notebook and pen ready to select and write down the database root password. It is unsafe to install the database (MySQL) with the default settings which includes a blank root password. The root user has access to all functions in the server. Therefore, anyone gaining root access to your server locally or over your network can make any change they desire on your server. It is important to select and write down a root password before you start the installation. Of course passwords should be secured to prevent compromising your servers.

TheWAMP can be placed online for everyone to access. Server administration will be covered in Topic 9. For now, explore the WAMP server by clicking on the WAMP server icon in your system try at the right end of your task bar.

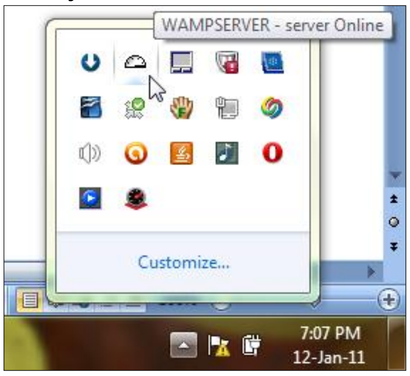


Figure 1: WAMP server icon

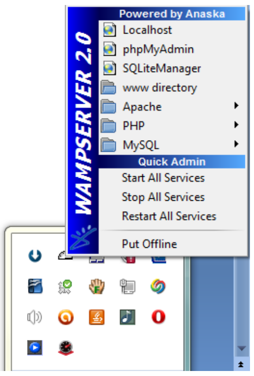


Figure 2: WAMP administration interface

The WAMP server administration interface provides you access to the configuration files for Apache, MySQl and PHP. You can also start and stop these servers. If you make a change to Apache or PHP, it is important to restart Apache.

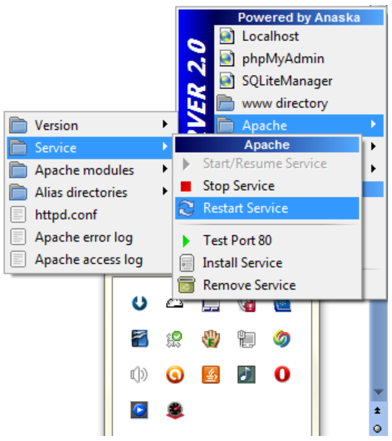


Figure 3: Restarting Apache [Apache->Service->Restart]

PHP Syntax

Variables, Comments and Identifiers

PHP variable names in PHP are case-sensitive but function names are not. White space between PHP keywords, variables and other tokens is ignored including spaces, tabs and newlines. Like Java PHP statements are terminated by a mandatory semicolon. Variables are marked with a “$” sign therefore $name and $dbase are PHP variables. Except for the $ sign, the same rules used in Java and Javascript for variable names apply.

Like Javascript, PHP variables have implicit data types depending on what is stored. These implicit data types include integer, floating point numbers, strings, and Booleans. There is no syntax for declaring variables without initializing them. Hence variables are declared as follows:

```
$name = "John";
$length1 = 123.345;
```

Strings can be enclosed in single or double quotes.

Long strings can be assigned using the *heredoc* syntax. Here is an example:

```
$example = <<<NKR
      There is no syntax for declaring variables without
      initializing them. Hence variables are declared as follows:
      $name = "John";
      $length1 = 123.345;
NKR;
```

The beginning of the text is marked by the chevron sign (<<<) with a unique character marker ('NKR' in this case). The end of the string is marked only with the unique marker.

PHP ignores whitespace between tokens. You can use spaces, tabs, and newlines to format and indent your code to make it more readable. PHP statements are terminated by semicolons.

There are three types of comments in PHP:

```
/* C style comments */
// C++ style comments
# Bourne shell style comments
```

The C++ and Bourne shell-style comments can be inserted anywhere in your code. Everything from the comment characters to the end of the line is ignored. In the C-style comment, PHP ignore everything from the start of the comment until the end-comment characters. Hence, this style of comment can span multiple lines.

Identifiers

Identifiers are names used to name variables, classes, constants, functions and formal parameters. The first character of an identifier must be either an ASCII **letter** (uppercase or lowercase), the underscore character (_), or any of the characters between ASCII 0x7F and ASCII 0xFF. After the initial character, these characters and the digits 0-9 are permitted.

Expressions and Operators

- PHP expressions are similar to Java and Javascript expressions. PHP's syntax is borrowed heavily from C and hence, is very similar to Java syntax. The same common operators used in Javascript is used in PHP. The following table shows the PHP operators in order of precedence and also shows associativity. Associativity is the order in which operators are evaluated if more than one operator at the same precedence level occur in the same expression. For example, division and multiplication are at the same order of precedence, in the expression x=2 + 4*6/7 the associativity rule for division and multiplication tells us that evaluation should be from left to right meaning that we will evaluate the operators starting from the left hand side. In this case, the multiplication will be done first (4*6) then the division (24/7).

Table 1: PHP Operators

Name	Operators	Associativity
Unary	`! ++ --`	Right to left

Multiplicative	`* / %`	Left to right
Additive	`+ -`	Left to right
Relational	`< <= > >=`	Left to right
Equality	`== !=`	Left to right
Logical AND	`&&`	Left to right
Logical OR	`||`	Left to right
Conditional	`?:`	Right to left
Assignment	`= += -= *= /= %=`	Right to left

Control Structures
The control structures for PHP are identical to those for Javascript. We looked at the repetition (for loop, while loop, do loop) and selection structures (if/else, switch) in Topic 4. The only additional repetition structure is the "foreach" structure used to loop through members of an array. The structure of foreach blocks is illustrated below:

```
foreach(array_name as $value) {
  statements
}
```

This loops through the array called *array_name* and assigns each value of the array to `$value` in turn. You can also get the key for each element with the following syntax:

```
foreach(array_name as $key=>$value) {
  statements
}
```

In the next section, we shall go a little deeper into this foreach statement to allow you to apply it with each when processing arrays.

Arrays and Strings
In PHP, variables are declared and initialised. Arrays are also treated in the same way. The following example shows how arrays can be created PHP:

```
$processors[0]="i7";
$processors[1]="Dual Core";
$processors[2]="Core2Dual";
$processors[3]="QuadCore";
```

Another way to create an array is as follows:

```
$x = array(1, 2, 3, 4, 5);
```

These arrays use an array index to identify and access individual members. They are called *numeric* arrays. However, it is also possible to use *associative* arrays in PHP. An associative array is one that uses strings to label the items in the array. The following is an example of how to create such an array:

```
$speed['i3'] = "3.2";
$speed['i5'] = "3.5";
$speed['i7'] = "3.6";
$speed['i9'] = "3.7";
```

Thereafter, you can access the array item through its name as follows:

```
echo $speed['i9'];  //echo is equivalent to document.write() in
Javascript
```

Associative arrays are useful in certain applications. You can use a foreach loop to access the array items as follows:

All arrays in PHP can be traversed safely with the following mechanism:

```
foreach($speed as $key=>$value) {
  echo "speed[$key]=$value<br>\n";
}
```

This loop will assign the "key" (labels) values of the array to the $key variable and the actual values to the $value variable then print the string "speed[xx]=yy" but insert the key values at xx and the actual item values at yy producing the following output:

```
speed[i3]=3.2
speed[i5]=3.5
speed[i7]=3.6
speed[i9]=3.7
```

Note that the names used to label the variables $key and $value have no special meaning. We could use $nakuru and $addis instead. This is the most common way to loop through each element of an array, whether it is a linear or an associative array. The foreach structure loops until the end of the array hence we do not need to know the size of the array to use this structure.

The general structure of a foreach loop is: **`foreach (array as value) { task to do; }`**
The following is an example of its use in a numeric array (one that uses numeric index):

```
 $x = array(1, 2, 3, 4, 5);

 foreach ($x as $y)
 {
 print $y . " "; //echo and print are almost identical
commands
 }
```

```
Output: 1 2 3 4 5
Dot (.) is a concatenation operator in PHP
```

Strings
In PHP strings are quoted with "double" or 'single' quotes. String concatenation is accomplished using the dot (.) operator:

```
<?php
$t1="Hello World!";
$t2="It is 2011!";
```

```
echo $t1 . " " . $t2;
?>
```

The output of the code above will be:

```
Hello World! It is 2011!
```

PHP has many string functions. Some references are provided at the end of this topic that describe these functions. Four useful functions are

- strtolower() Converts a string to lowercase letters
- strtoupper() Converts a string to uppercase letters
- strlen() Returns a string
- strpos() Returns the position of a string or character within another string.

Table 2: String function examples

Example	Output	Explanation
`<?php` `echo strlen("The weather is fine!");` `?>`	20	There are20 characters including the spaces and the exclamation mark. Useful in processing strings when searching for a character. Loop counter is often the string length as illustrated below
`<?php` `echo strpos("The weather is fine!","weather");` `?>`	4	We count position from 0 (zero) hence “weather” begings from position 4 in the string.
`<?php` `echo strtoupper("The weather is fine!");` `?>`	THE WEATHER IS FINE!	
`<?php` `echo strtolower("THE WEATHER IS FINE!");` `?>`	the weather is fine!	

Print and Echo

The two keywords used in PHP for output are “echo” and “print”. The following are examples of their use:

```
print "Hello World! <br />";
echo "Hello World! <br />";
```

The two output statements above are identical in the way they work. They will both produce the word “Hello Word” on your web page with a line break in between.

```
print ("Hello World! <br />"); //behaves like a function
echo ("Hello World! <br />"); //behaves like a function
echo "Hello" , "World!" , "<br />"; //Can print a list of values
```

Variables may be interspersed with string text as shown below. This permits us to dynamically change the content of a web page For example, if $date contained today’s date, the statement: echo “Today’s date is $date”; would print the current date rather a fixed date.

```
$name = “Annan”;
echo (“Hello Mr. $name”); //Will print “Hello Mr. Annan”
echo (“Today’s date is” . date(1));//Prints today’s date using one
available format
```

Functions
As seen earlier while learning Javascript, a function is a named groups of statements that typically carry out a specific task or compute a result. They are normally designed to accept parameters and return a value. A function call is an expression that has a value; its value is the returned value from the function. PHP provides a large number of internal functions. PHP supports user-definable functions. PHP functions are similar to Javascript functions:

```
function sumOfTwoNumbers($a, $b,)
 {
    return $a + $b;
}
```

Note that, unlike in Java methods where the return type must be specified, PHP, like Javascript is not a strongly typed language and requires no explicit return type.

To call the function, state its name and provide the arguments. For example:

```
$c = sumOfTwoNumbers($a, $b,)
Or
$d = sumOfTwoNumbers(234.4, 456.6)
```

Variable Scope in Functions
Variables have scope. Scope refers to those parts of a program within which this variable can be accessed. PHP variables are accessible from all parts of the program except if they are is declared within a function. Such a varaible cannot be accessed from outside this function. However, there are cases when you want a variable inside a function to be accessible from outside that function. It is said to be a “local variable. We would then need to convert a local a variable into a global variable. The next section shows how this is done.

Global and Static variables
Global variables are variables defined in such a way that they can be accessed from any part of your program. The common way to define globals is to place the word “global” in from of the variable name during declaration as shown in the example below:

```
function sumOfTwoNumbers($a, $b)
 {
    global $c;
     $c = $a + $b;
    return $c;
}

  $a = 234;
  $b = 456;
  echo sumOfTwoNumbers($a, $b);
```

Static variables are special local variables (declared inside a function) that are able to maintain their value between function calls. This means that, if they acquire a value when the function is called, they maintain this value. Normally, a local variable exist only during a function call and disappears after that. When the same function is called again, the variable is recreated afresh.
To convert a local variable into a static variable, put the word “static” in from of it as follows:

```
static $counter = 0;
```

Static variables are usually used as counters. They remain local in scope.

Your user-defined function names should not conflict with PHP's predefined function names. If you get the following error, it means your function names conflict with one another:

```
Fatal error: Can't redeclare already declared function in filename
on line N
```

Function Libraries

One of the strengths of PHP is its extensive function library. PHP has a large number of functions available for use by programmers. This abundance of functions is one of the key reasons why PHP is popular as a web programming language. Some of these categories of functions available in PHP are:

- Array functions
- Calendar functions
- Date functions
- Directory functions
- Error functions
- Filesystem functions
- Filter functions
- FTP functions
- HTTP functions
- LibXML functions
- Mail functions
- Math functions
- Misc functions
- MySQL functions
- SimpleXML functions
- String functions
- XML Parser functions
- Zip functions
- GD Graphics library
- PEAR MDB2 database library.

As an example, the following script prints today's date using the specified format. You can investigate the intricacies of the date function at: http://www.php.net/manual/en/function.date.php The tutorials at the end of this topic also provider further information on PHP functions. Learning PHP is largely about learning how to effectively use PHP functions.

Table 3: Date() function example

Script	Output

`<?php`
`echo "This is a date example<br/>";`
`$date = date('l jS \of F Y');`
`echo ("Today's date is $date");//`
`?>`
	This is a date example
	Today's date is Wednesday 12th of January 2011

Simple PHP Programs

PHP is used principally to add dynamic content to web pages and implement web-based applications. One of the most frequent structures that you need to be able to create dynamically is an HTML table.

The common strategy for creating a table with PHP is to print the table header using a simple PHP print statement such as:

```
print " <table><tr><th> Semester</th><th>Code </th><th>
Title</th></tr>";
```

You can now add the table rows dynamically using a loop as follows:

```
while ($row = mysql_fetch_assoc($result)) {
        print "<tr><td>$row['sem']</td>";
        print "<td>$row['code']</td>";
        print "<td>$row['title']</td></tr>";
               }
```

Where the function mysql_fetch_assoc($result) fetches a row of data from a database and stores it in an **associative** array called $row. The individual array values can be accessed using the notation we discussed earlier in this topic. In this case, there are three items: $row['sem'], $row['code'], $row['title']. PHP has a similar function that does not use associative array to fetch data (mysql_fetch_array($result)). Using this function the loop above would change as shown below. Note the use of a concatenation operator. The results are identical to the example above though the appearance looks different. Different programmer have different copding preferences> PHP gives you the flexibility to make your chose of coding style. Sometimes one code format may perform less efficiently than another. See a good discussion of performance of PHP applications at: http://phplens.com/lens/php-book/optimizing-debugging-php.php

```
while ($row = mysql_fetch_array($result))
{
```

```
print("<tr><td>".$row[0]."</td><td>".$row[1]."</td><td>".$row[2]."</td
></tr>");
}
```

For each loop a new row is fetched and its values are placed in the three table cells in each row.
The loop automatically stops when there is no more data coming from the database. In the next section, we shall see how we can create a simple database and fetch data from the database into a dynamic HTML table.

After the table rows are created, we issue a print command to close the table:

```
print "</table>";
```

Quiz 1

1. Complete the following PHP code so that the print command prints out the contents of the array:
   ```
   $precious = array("Jasper", "Sapphire", "beryl", "topaz",
   "turquoise", "jacinth", "amethyst");

    foreach (___________________)
    {
    print ________________; //echo and print are almost identical
   commands
    }
   ```
2. A table has three columns (Course_Code, Course_Title, and Course_CF). Complete the following PHP script to print an HTML table based on the data from this table:
   ```
    while ($row = mysql_fetch_array($result))
   {

   print("_______________________________________________________");
   }
   ```

Creating a MySQL Database
Before we can create database publishing applications, it is essential to learn the basics of database implementation. In this course, we shall not cover the theory of database design but restrict ourselves to implementation of simple databases for use in our web applications.
Once MySQL is installed, we can create databases using a variety of strategies. There are three popular tools for administering MyQL that we can use to manage the database creation process. These are summarised in the table below:

Table 4: MySQL Administrator Tools

MySQL Console	phpMyAdmin	MySQL Administrator
This tools is built into MySQL and can be accessed through the WAMP administration interface described earlier in this	This tool is web-based hence provides remote access to a MySQL server. We can use it to upload data into a MySQL database, browse databases, add users and so on.	This is a Windows graphical user interface that provides access to your databases. It is more secure than the phpMyAdmin because it require user login and only

topic. It is a command-line interface. We shall discuss its use in some detail.	However, it is insecure because it saves the root password and can be used by an unauthorised person to access your databases.	provides access to the databases the user is authorised to access.

Expert users prefer command line tools. Beginners prefer web-based on GUI tools. In this Topic and in the following topic, we shall examine the use of the Mysql console. Figure 4 shows how to access the MySQL console from the WAMP administration interface. You will require the MyQL root password before you can access the console.

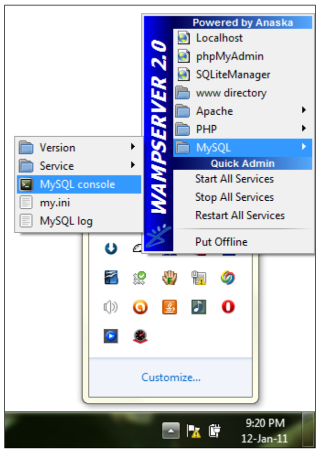
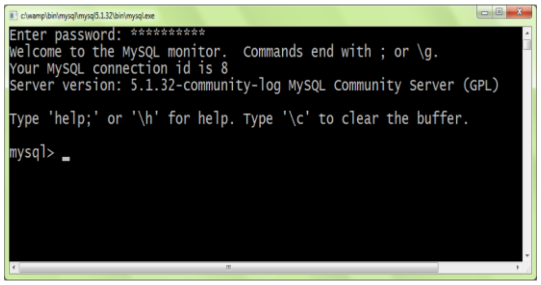


Figure 4: How to access MySQL console



```
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.1.32-community-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

Figure 5: MySQL console

The following instructions provide a brief introduction to the use of the console to create databases.

Creating a Database

The MySQL console normally requires a request in the following format to connect to a database:

```
mysql -h host -u user -p
```

You would then supply password from prompt

host is the IP number or domain name of computer hosting MySQL
user is your user name

However, in the MySQL console built into the WAMP server administration interface, it is assumed you are the root user and requests only for the root password.

Figure 5 shows the console ready for user input with the characteristic prompt (mysql>). The following table is a summary of useful commands:

Table 5: Useful MySQL commands

MySQL console command	Explanations
show databases;	List existing databases (note semicolon)
use phonebook;	Use a specific database. In this case the database is "phonebook"
show tables:	Show the tables that make up the database
describe table_name;	Describe the structure of a table whose name is "table_name"

create database participants;	Create a database called “participants”
`CREATE TABLE Students ( Name VARCHAR(40) NOT NULL, ID VARCHAR(40) NOT NULL, PRIMARY KEY (ID) );`	Create a table called “Students” with the structure shown. It has two columns (name and ID) with **varchar** data types (variable character) with a width of 40 characters each. The primary key is ID. “NOT NULL” means that the field can not be empty.
`LOAD DATA INFILE "courses1.csv" INTO TABLE comp FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n' ;`	This is a very useful command to import data into MySQL from a local file (`"courses1.csv"`). The data should be in CSV format and the .CSV file should be placed in the folder containing the database. In a typical WAMP installation, this is located at: **C:\wamp\bin\mysql\mysql5.1.32\data\example1** Where “example1” is the name if the database. Your database location may be different especially if your version of MySQl is not 5.1.23

When typing commands into the MySQL console, the command can span multiple lines and will not terminate until you enter the semicolon and press ENETER. To create a database and import data into the database, use the following procedure:

1. Go into the console and create a database using the command “`create database_name`” where database_name is the name of the database you want to create.
2. Design a simple table to hold the data you desire. For example, a phone book may have the following structure:

Table 6: Table structure

Name	Phone (Primary key)
VARCHAR(45)	VARCHAR(16)
LEONARD WEKU KUCHA	0723335631
ANNE BAA	0721966978
CHEROKA CAROLYNE	0723090708
CATALY KOECH	0721655252
LEAH CHEMO	07298655262

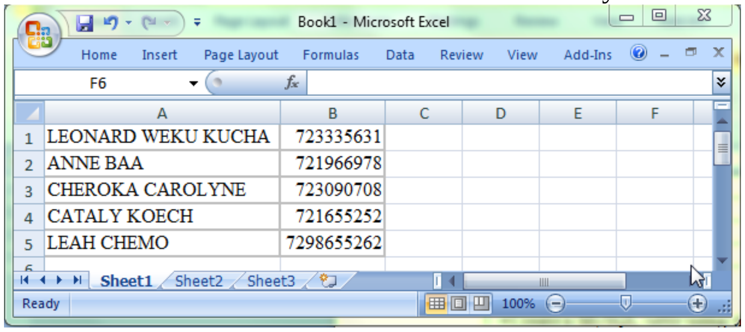
3. Create a MySQL table using the command:

```
CREATE TABLE phones (
Name VARCHAR(45) NOT NULL,
Phone VARCHAR(45) NOT NULL,
PRIMARY KEY (Phone)
      );
```

4. All you now need is to import data into the database.

Importing Data using LOAD INFILE

Before data is imported, it must exist. The easiest approach to creating the data is to enter the data into an Excel spreadsheet then save it as a .CSV file stored in the same folder as your database.



	A	B
1	LEONARD WEKU KUCHA	723335631
2	ANNE BAA	721966978
3	CHEROKA CAROLYNE	723090708
4	CATALY KOECH	721655252
5	LEAH CHEMO	7298655262

Figure 6: Excel data

- Save the Excel worksheet as a .csv file in the same location as the database:

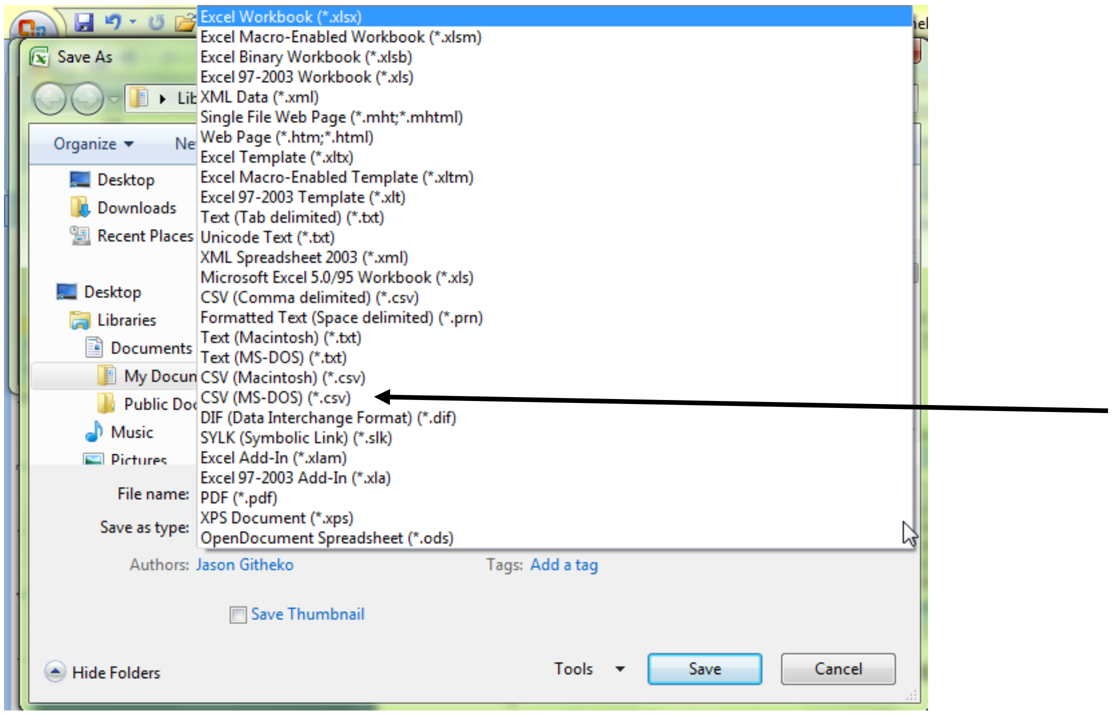
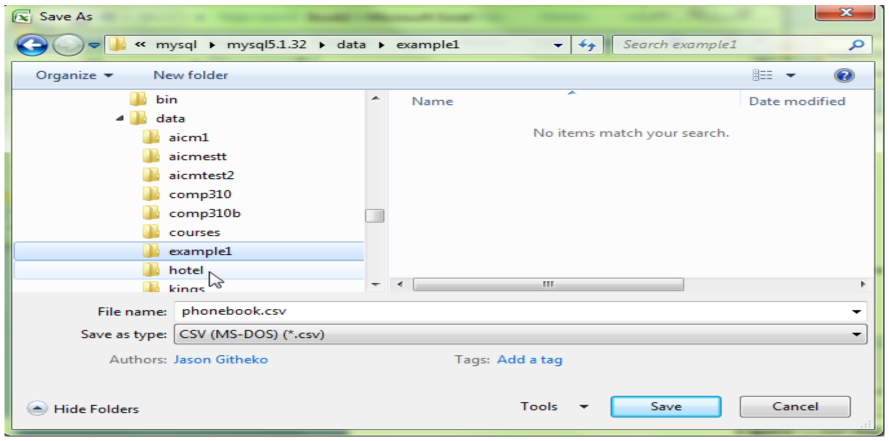

Save As
Organize
Desktop
Downloads
Recent Places
Desktop
Libraries
Documents
My Docun
Public Doc
Music
Pictures
File name:
Save as type:
Excel Workbook (*.xlsx)
Excel Macro-Enabled Workbook (*.xlsm)
Excel Binary Workbook (*.xlsb)
Excel 97-2003 Workbook (*.xls)
XML Data (*.xml)
Single File Web Page (*.mht;*.mhtml)
Web Page (*.htm;*.html)
Excel Template (*.xltx)
Excel Macro-Enabled Template (*.xltm)
Excel 97-2003 Template (*.xlt)
Text (Tab delimited) (*.txt)
Unicode Text (*.txt)
XML Spreadsheet 2003 (*.xml)
Microsoft Excel 5.0/95 Workbook (*.xls)
CSV (Comma delimited) (*.csv)
Formatted Text (Space delimited) (*.prn)
Text (Macintosh) (*.txt)
Text (MS-DOS) (*.txt)
CSV (Macintosh) (*.csv)
CSV (MS-DOS) (*.csv)
DIF (Data Interchange Format) (*.dif)
SYLK (Symbolic Link) (*.slk)
Excel Add-In (*.xlam)
Excel 97-2003 Add-In (*.xla)
PDF (*.pdf)
XPS Document (*.xps)
OpenDocument Spreadsheet (*.ods)
Authors: Jason Githeko
Tags: Add a tag
Save Thumbnail
Hide Folders
Tools
Save
Cancel

Figure 7: Saving as .csv in Excel

When saving the Excel file, select the .csv file format. Different versions of Excel use a different method to select this file format.

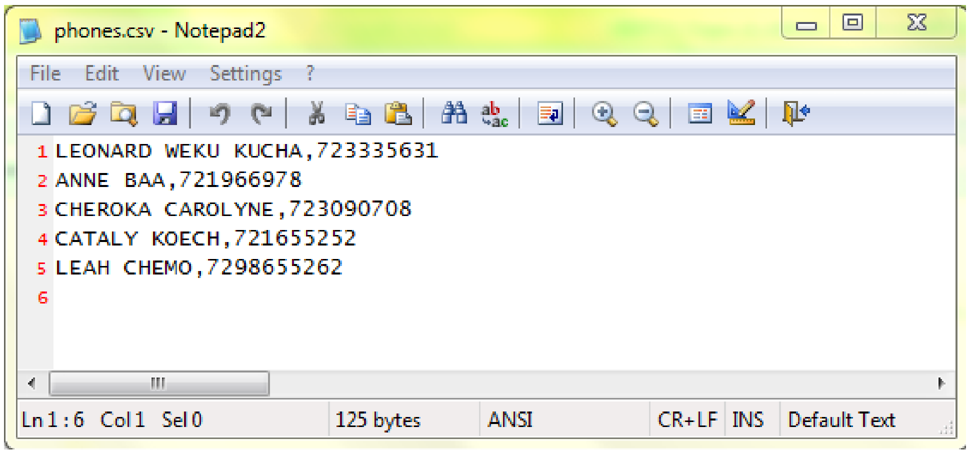
In the example shown in Figure 8, the file is to be saved in the following folder:
C:\wamp\bin\mysql\mysql5.1.32\data\example1



File name: phonebook.csv
Save as type: CSV (MS-DOS) (*.csv)
Authors: Jason Githeko
Tags: Add a tag

Figure 9: Saving the .csv file

If all is well, then, in the folder **C:\wamp\bin\mysql\mysql5.1.32\data\example1** you should find your data file in .csv format. Open this file with a plain text editor to examine the structure. MS Excel sometimes inserts quotation marks that can confuse MySQL. The above example is shown in Figure 10 in CSV format.



```
LEONARD WEKU KUCHA,723335631
ANNE BAA,721966978
CHEROKA CAROLYNE,723090708
CATALY KOECH,721655252
LEAH CHEMO,7298655262
```

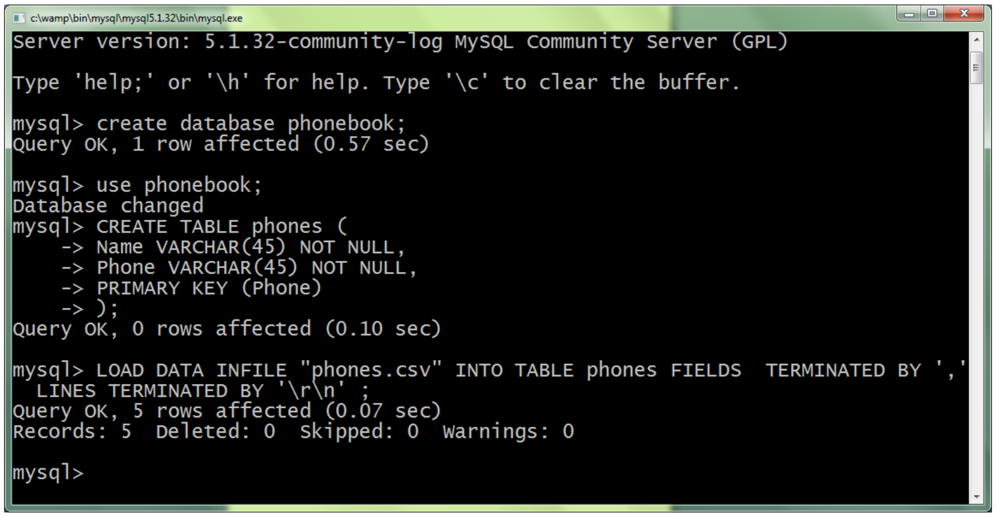
Ln 1 : 6 Col 1 Sel 0 | 125 bytes | ANSI | CR+LF | INS | Default Text

Figure 10: Contents of CSV file

The LOAD INFILE command looks rather complex but if you use the SCV format described here, all you need to do is to indicate the name of the data file and the name of the table into which it is to be imported. In our example, the LOAD command will look like this:

```
LOAD DATA INFILE "phones.csv" INTO TABLE phones FIELDS  TERMINATED BY
','  LINES TERMINATED BY '\r\n' ;
```

This command can now be entered into the MySQL console as shown in Figure 11. Figure 11 shows all the MySQL commands used to create and populate the database.



```
Server version: 5.1.32-community-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database phonebook;
Query OK, 1 row affected (0.57 sec)

mysql> use phonebook;
Database changed
mysql> CREATE TABLE phones (
    -> Name VARCHAR(45) NOT NULL,
    -> Phone VARCHAR(45) NOT NULL,
    -> PRIMARY KEY (Phone)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> LOAD DATA INFILE "phones.csv" INTO TABLE phones FIELDS  TERMINATED BY ','
  LINES TERMINATED BY '\r\n' ;
Query OK, 5 rows affected (0.07 sec)
Records: 5  Deleted: 0  Skipped: 0  Warnings: 0

mysql>
```

Figure 11: MySQl commands to create and populate a database

Publishing Data with PHP

In Topic 7, we shall go into greater detail with regard to publishing data using PHP. In this section, a complete example of how to fetch data from a live database and publish it on a webpage is shown. This technique requires that you refresh the web page to update the data displayed. However, AJAX permits you to create web pages where data is automatically refreshed without reloading the webpage.

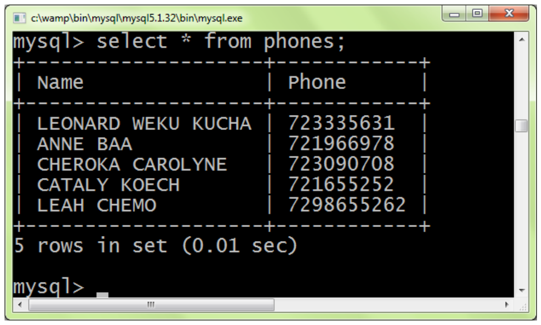
Most database systems today are relational database management systems that use the Structured Query Language (SQL) to manipulate data and search for data. A detailed discussion of SQL is beyond the scope of this course but we shall introduce a few SQL commands to allow us build simple PHP applications. The most useful SQL command is the SELECT query command. It is used to query a database to select records according to some criteria. In its simplest form, a select query is used to select all data from a table as follows:

```
SELECT *
FROM phones;
```

Quiz 2

1. Write a CREATE SQL statement to create a database table that lists people's names and email addresses. Name the table "email"
2. Write a SELECT query that returns all the data from the table created above.
3. You are to import data into the table created in Question 1. Write a LOAD DATA statement to load this data from a CSV file names "data.csv."

The semicolon is required in MySQL. Other RDBMS may not require it.



```
mysql> select * from phones;
+---------------------+------------+
| Name                | Phone      |
+---------------------+------------+
| LEONARD WEKU KUCHA  | 723335631  |
| ANNE BAA            | 721966978  |
| CHEROKA CAROLYNE    | 723090708  |
| CATALY KOECH        | 721655252  |
| LEAH CHEMO          | 7298655262 |
+---------------------+------------+
5 rows in set (0.01 sec)

mysql> _
```

Figure 12: Output from SELECT query

To simplify your application development process, follow the procedure describe in the following diagram (Figure 13):

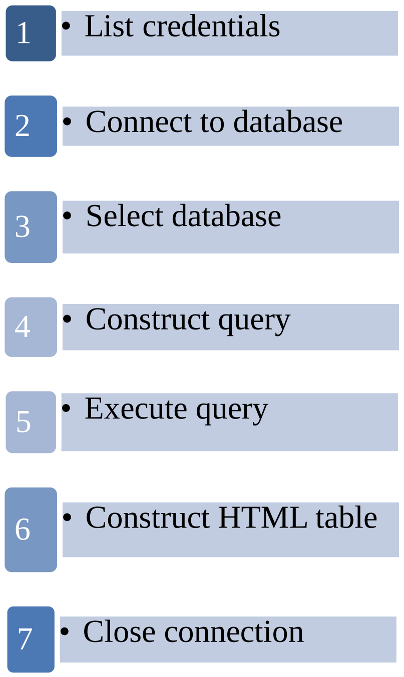


Figure 13: Simplified data publishing process

The following example is well documented showing the MySQL-specific functions in PHP that are used for:

- Connecting to a database – `mysql_connect('localhost', $username, $passwd);`
- Selecting a database - `mysql_select_db($database);`
- Executing a query string - `mysql_query ($dbquery, $dbhandle);`
- Closing a connection - `msql_close($dbhandle);`

Earlier we saw how to construct an HTML table using PHP code.

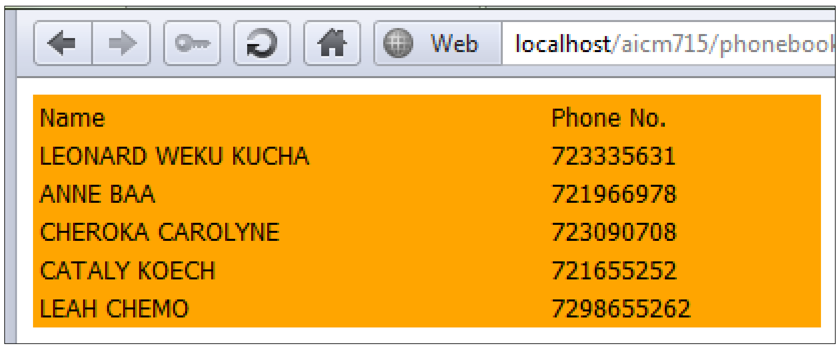
The complete example is shown in Table 7

Table 7: Complete database publishing example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html><head><title>Data Publishing Example</title></head><body>
<?php
     $username="root"; //user name
     $passwd="19141976"; //user password
     $database="phonebook"; //database name
        $dbhandle = mysql_connect('localhost',  $username, $passwd);
//connect to database

        if ($dbhandle == false)
        {
                 die  ("Unable to connect to MySQL Database<br>");
//If connection fails, show error message
        }

        $db = mysql_select_db($database); //select the database to use
        if ($db == false)
        {
                 die  ("Unable to Select MySQL Database<br>"); //if
database selection fails, issue error msg
       }

        $dbquery = "SELECT * FROM phones "; //Construct SQL query
        $dbresult = mysql_query ($dbquery, $dbhandle); //Execute the
query and store results
        if ($dbresult == false)
        {
                die  ("Unable to perform query<br>"); //if query
fails, issue error msg
        }
        //Construct HTML table header
echo <<< TAB
<table width="30%"style='background: orange; font-family: tahoma;
font-size: 10pt;'><tr><td>Name</td><td>Phone No.</td></tr>
TAB;
        while ($dbrow = mysql_fetch_row ($dbresult))  //Fetch a row at
a time

        {
                print("<tr><td>$dbrow[0]</td>"); // Construct table
rows
                print("<td>$dbrow[1]</td></tr>");

        }
echo " </table> "; //Close table

mysql_close($dbhandle); //Close database connection
?>

</body>
</html>
```

The following is the output of this script on a browser:



Name	Phone No.
LEONARD WEKU KUCHA	723335631
ANNE BAA	721966978
CHEROKA CAROLYNE	723090708
CATALY KOECH	721655252
LEAH CHEMO	7298655262

Figure 14: Data published on a web page

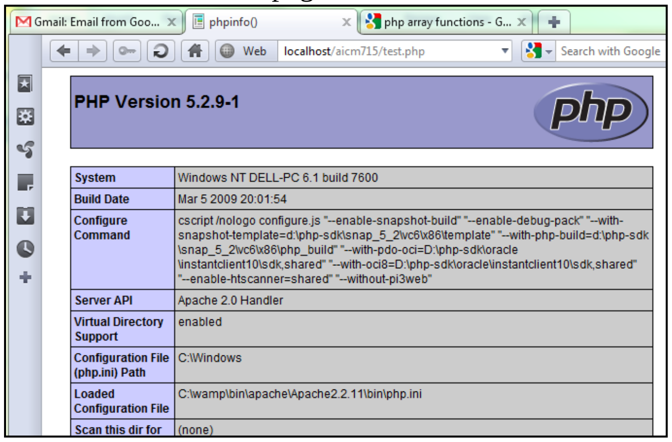
Learning Activities

Exercise 1

i. Install WAMP server
ii. Copy the following program and place it in your web server document root directory (Usually c:/wamp/www). You can call the file "test.php".

```
<?php
phpinfo();
?>
```

iii. Load the file with a browser by pointing to: http://localhost/test.php
iv. You should see a PHP information page that looks as follows:



PHP Version 5.2.9-1

System	Windows NT DELL-PC 6.1 build 7600
Build Date	Mar 5 2009 20:01:54
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-snapshot-template=d:\php-sdk\snap_5_2\vc6\x86\template" "--with-php-build=d:\php-sdk\snap_5_2\vc6\x86\php_build" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--enable-htscanner=shared" "--without-pi3web"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\wamp\bin\apache\Apache2.2.11\bin\php.ini
Scan this dir for	(none)

v. Resolve any issues that prevent the PHP script from running.

Exercise 2

i. Copy Example 1 into a text editor such as Scite or Notepad2 or Notepad++
ii. Save the file as Example1.php in your document root
iii. Open the file in your browser by pointing to http://localhost/Example1.php

Exercise 3

Write a PHP script that prints the Fibonacci series (1, 1, 2, 3, 5, 8, 13, ...)on the browser window vertically starting from the first member to the 10th member.

Exercise 4

Using a PHP script, create a Square root table of numbers from 1 to 100 using the following format*:*

Number	*Square root*
1	*1*
2	*1.4142135623730951*
3	*1.7320508075688772*

The PHP function that computes square roots is sqrt(). Your script should construct the table dynamically.

Exercise 5

Modify your solution to Exercise 4 so that, instead of square roots, the script should print a list of students and their email addresses. Use dummy student data or the current class list if available.

Forum: Participate in the PHP/MySQL programming discussion forum.

Self Study 1: Study the operation of the following SQL commands:

- `UPDATE abc SET xyz`
- `DROP TABLE abc`
- `SELECT * FROM efg WHERE ijk = “stu“`
- `INSERT INTO shop VALUES (1,'A',3.45),…….;`

Self Study 2: Study the MySQL tutorial from http://   and identify the main data types used in MySQL.

Assignment 1

Create your own database and publish it on a webpage. Prepare a class demonstration of your work during the next practical session.

Summary of Topic

In this topic we looked at the architecture of web applications using PHP/MySQL as an example. PHP syntax was introduced. Several examples were provided on how PHP scripts are incorporated into web pages to add dynamic server-side effects. In particular, we examined in detail the process of database publishing using the MySQL database management software. We learned how to install and Apache, PHP and MySQL as a single package (WAMP) and how to create a database and populate it with data using the MySQL console command LOAD INFILE. A structured procedure for data publishing was proposed that simplified the task of publishing live data on a web page. In the next topic, we shall examine database publishing in greater detail and look at security and application performance issues.

Assessment

Please find the review **Quiz 4** online at http://learn.egerton.ac.ke

Further Reading Materials

Text Books:

Harvey M. Deitel and Paul J. Deitel (2007). *Internet & World Wide Web: How to Program* (4th Edition). Prentice Hall(ISBN-10: 0131752421, ISBN-13: 978-0131752429)

Lerdorf, R and Tatroe, K. (2002). Programming PHP. O’Reilly.

Sklar, D. (2004). Learning PHP 5: A Pain-Free Introduction to Building Interactive Web. O'Reilly Media

Web-Based PHP Tutorials

- http://www.tutorialspoint.com/php/index.htm
- http://www.w3schools.com
- http://www.tizag.com/htmlT/
- MySQL Manual: http://dev.mysql.com/doc/ [Each major version has a separate manual including a tutorial]
- MySQL 5.5 Tutorial: http://dev.mysql.com/doc/refman/5.5/en/tutorial.html

PHP String Functions

- http://w3schools.com/PHP/php_ref_string.asp

Online PHP Books

- Gutmans,A., Bakken,S. and Rethans, D. (2004). PHP 5 Power Programming. Prentice Hall [http://www.informit.com/content/images/013147149X/downloads/013147149X_book.pdf]
- Smyth, N. (2007).PHP Essentials http://www.techotopia.com/index.php/PHP_Essentials
- Hudson, P. (2003 - 2009). Practical PHP Programming. [http://www.tuxradar.com/practicalphp]
- Wikibooks. PHP Programming. http://en.wikibooks.org/wiki/PHP_Programming