Developing Web Applications with PHP

Raghuram.N
Free Software Movement Karnataka

www.fsmk.org

Content

- Introduction
- PHP on Linux
- PHP Language Basics
- PHP with Web Forms
- Functions
- Strings
- Arrays
- Advanced PHP

Introduction

What is PHP?

- PHP – Hypertext Processor
- Embedded Scripting language for HTML
- Combines elements of Perl, C and Java
- Creates Dynamic Web Page
- Server Side Scripting
- Command line scripting
- It generates XML doc, PDF files, Flash animation

Introduction

History of PHP

- Created by Rasmus Lerdorf in 1995 for resume tracking.
- Originally a set of Perl scripts known as the “Personal Home Page”.
- Zeev Suraski and Andi Gutmans released Zend Engine in 1998. It supports ODBC database, multiple platform.

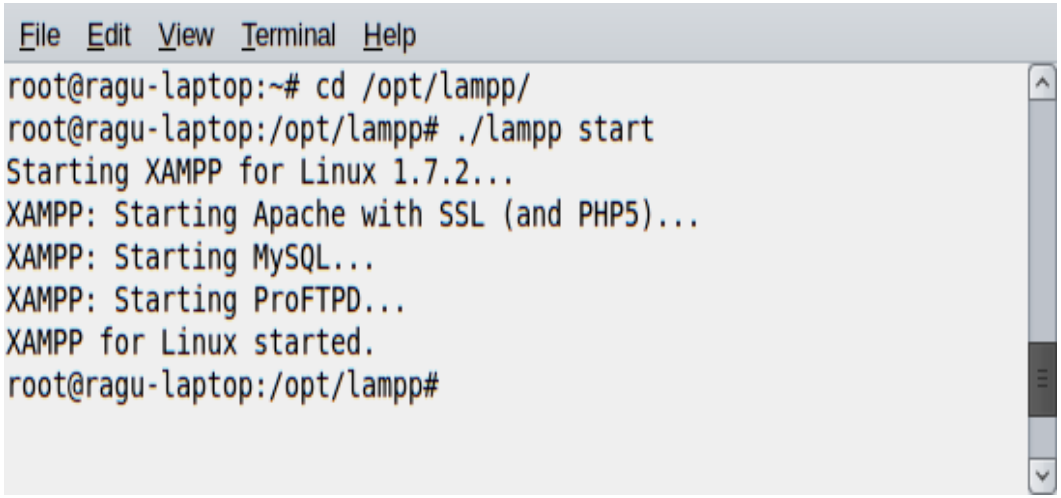
Introduction

Free Software – Source code is available, User can read and modify the source code – PHP License
Platform Independent – Runs on Linux, Unix, Solaris, Mac OS X and Windows
Web Server: Apache, Netscape Enterprise Server and Microsoft IIS
Supported Database: ODBC, Oracle, MYSQL, SQLite, PostgreSQL and so on.

PHP on Linux

- Download xamp server - xampp-linux-1.7.2.tar.gz
- Extract this file into **/opt/**

Start xamp Server:



PHP Language Basics

The Script Tags:

- <?php //Code ?> => XML Style
- <? //Code ?> => SGML Style
- <% //Code %> => ASP Style
- <script language=”php”> //Code </script> => Script Style

Eg:-

```
<?php
echo “Hello AIT”;
?>
```

PHP Language Basics

Data Types:

- Strings
- Integers
- Float
- Boolean
- Array
- Object
- Resource

PHP Language Basics

Variables:

- Simillar to C, Java variables
- Its an identifier which points to a memory location
- Variable start with $ - Dollar sign
- Its Losely/Weakly Typed Language

Variables Declaration in C, Java:

Int a=10;
float f=5.2;

Variables Declaration in PHP:

$a=10;
$f=5.2;

PHP Language Basics

Variables - Rules:

Special Characters are not allowed

*, +, #, @, ^, ! and so on

Invalid Variables:

$city*

$address+

$emp#name

$emp@emailid

Valid Variables:

$city

$address1

$emp_name

Constants:

Value does not change throughout the execution

Syntax

define(“ConstName”,value);

Eg:

define(“THE_YEAR”,2010);

echo “The year: “.THE_YEAR;

echo “The year: “.the_year;

PHP Language Basics

Operators:

- Arithmetic Operators
- Increment/Decrement Operators
- Comparison Operators
- Bitwise Operators
- Logical Operators
- Casting Operators
- Combined Assignment Operators
- Conditional Operators

PHP Language Basics

Arithmetic Operators

Si No	Operation	Operator
1	Additional	+
2	Subtraction	-
3	Multiplication	*
4	Division	/
5	Modulus	%

PHP Language Basics

Increment/Decrement Operators

Si No	Operation	Operator
1	Increment	++
2	Decrement	--

PHP Language Basics

Comparison Operators

Si No	Operation	Operator
1	Value Equality	==
2	Value and Type Equality	===
3	Not Equal	!= and <>
4	Type and Value Inequal	!==
5	Greater than	>
6	Greater than or equal	>=
7	Less than	<
8	Less than or equal	<=

PHP Language Basics

Bitwise Operators

Si No	Operation	Operator
1	Bitwise Negation	~
2	Bitwise AND	&
3	Bitwise OR	|
4	Bitwise XOR	^
5	Left Shift	<<
6	Right Shift	>>

PHP Language Basics

Logical Operators

Si No	Operation	Operator
1	Logical AND	&&
2	Logical OR	||
3	Logical XOR	xor
4	Logical Negation	!

PHP Language Basics

Casting Operators

Si No	Operation	Operator
1	To Integer	(int)$a
2	To Float	(float)
3	To String	(string)
4	To Boolean	(bool)
5	To Array	(array)
6	To Object	(object)

PHP Language Basics

Combined Assignment Operators

Si No	Operation	Operator
1	Assignment	=
2	Plus Equal	+=
3	Minus Equal	-=
4	Multiply Equal	*=
5	Divide Equal	/=
6	Modulus Equal	%=
7	Bitwise AND Equal	&=
8	Bitwise OR Equal	|=
9	Bitwise XOR Equal	^=
10	Concatenate Equal	.=

PHP Language Basics

Conditional Operators

Si No	Operation	Operator
1	Conditional	cond?true:false

Eg:

```
$a=10;
$b=5;
$big=($a>$b)?$a:$b;
```

PHP Language Basics

Flow Controls

- PHP supports C, Java's flow controls(Conditional Statements)
- Executes a statement depending on some condition

if

if/else

switch

PHP Language Basics

Flow Controls - if

Syntax

```
if(exp)
    true stmt;
```

Eg:-

```
<?php
$stu_name="Raghu";
if($stu_name=="Raghu")
  echo "Welcome $stu_name<br>";
?>
```

Output:

Welcome Raghu

PHP Language Basics

Flow Controls - if/else

```
if(exp)
   //true stmt;
 else
     //else stmt;
```

Eg:-

```
<?php
$stu_name="Ram";
if($stu_name=="Ram")
     echo "Welcome $stu_name<br>";
else
     echo "Access Denied<br>";
?>
```

PHP Language Basics

Flow Controls - switch

Syntax	Eg

switch(expression)
{
case 1:
First case stmt
break;
case 2:
Second case stmt
break;
default:
default stmt
break;
}
	<?php
	$name=”Ram”;
	switch($name)
	{
	case 'Raghu':
	echo “Hi Raghu”;
	break;
	case 'Ram':
	echo “Hi Ram”;
	break;
	default:
	echo “Access Denied”;
	break;
	}

PHP Language Basics

Loops

- Repeats block of statement
  - for
  - while
  - do while
  - foreach

PHP Language Basics

Loops - for

Syntax1	Syntax2	Eg

for(initialization;Condition
;Increment)
{
Stmt
}
	for(initialization;Condition;
	Increment):
	Stmt
	endfor;
		for($i=0;$i<10;$i++)
		{
		echo $i;
		}

PHP Language Basics

Loops - while

Syntax1	Syntax2	Eg

Initialization
while(Condition)
{
Stmt
Increment/Decrement
}
	Initialization
	while(Condition):
	Stmt
	Increment/Decrement
	endwhile;
		$i=0;
		while($i<10)
		{
		echo $i;
		$i++;
		}

PHP Language Basics

Loops – do while

Syntax	Eg
Initialization do { Stmt Increment/Decrement }while(Condition);	$i=0; do { echo $i; $i++; }while($i<10);

PHP Language Basics

Loops – foreach

- Its used with arrays

Syntax1	Syntax2	Eg

foreach($arrayName as $element)
{
Stmt
}
	foreach($array as $elem):
	Stmt
	endforeach;
		foreach($array as $elem)
		{
		echo “Element: $elem”;
		}

PHP with Web Forms

Gives global variable to access web form element

Global Variable

- $_GET – get method
- $_POST – post method
- $_REQUEST – get/post method

Access Textfield

```
$Name=$_POST[“txtName”]; //txtName is name of Textfield
```

Access Radio Button

```
$gender=$_POST[“rdGend”]; //rdGend is name of Radio button groups
```

Access Dropdown list

```
foreach($_POST[“ddlCourse”] as $val)
{
if($val!=””)
$ddlVal=$val;
}
```

Function

- Block of code that performs certain task
- It can be called more than one time
- Its compiled only one time.
- Allows user to reuse the code
- Input – zero or more parameters

Types of Function

- Built-in/Pre-defined function
- User-defined function

Function

Types of Function

- Built-in/Pre-defined function
- User-defined function

Pre-defined Function

- The function which is already defined in language

Eg:-

$len=strlen(“Raghu”);

User-defined Function

- The function which is defined by user

Function

Define a Function

- Define a function with keyword 'function'

Eg:-

```
function some_function($arg1,$arg2)
{       //Some Code     }
function other_function()
{       //Some Code     }
```

Calling a Function

- Call the function with function name
- Pass parameters if function definition has parameters

Eg:-

```
some_function($param1,$param2);
other_function();
```

Function

Returning Values from User-Defined Functions

- A function can return a value/output using keyword '**return**'
- The statment return stops execution of the function and execution control back to the function calling point.
- Return value can be hard-coded
- Return value can be return value of another function

Eg:-

```
function arithmetic_Sum($val1,$val2)
{
        $result=$val1+$val2;   //return ($val1+$val2);
        return $result;
}
```

Strings

Strings

- Its collection of characters
- PHP provides many function to manipulate string

String Formatting

- printf()
- sprintf()

Strings

printf()

- Similar to printf() function in C
- It requires an argument - “Format Control String” or “Type Specifier”.

Type Specifier

Specifier	Description
d	Display argument as a decimal number
b	Display an integer as a binary number
c	Display an integer as ASCII equivalent
f	Display an integer as a floating-point number (double)
o	Display an integer as an octal number (base 8)
s	Display argument as a string
x	Display an integer as a lowercase hexadecimal number (base 16)
X	Display an integer as a uppercase hexadecimal number (base 16)

Strings

Printf() - Eg:

Prog	Output
<?php $no=490; echo "Type of variable: $ty "; printf("Decimal: %d",$no); printf(" Binary: %b",$no); printf(" Double: %f",$no); printf(" Octal: %o",$no); printf(" String: %s",$no); printf(" Hexadecimal(Lower): %x",$no); printf(" Hexadecimal(Upper): %X",$no); ?>	Decimal: 490 Binary: 111101010 Double: 490.000000 Octal: 752 String: 490 Hexadecimal(Lower): 1ea Hexadecimal(Upper): 1EA

Strings

String Manipulation:

strlen()	str_ireplace()
strpos()	str_word_count()
stripos()	trim()
strtoupper()	ltrim()
strtolower()	rtrim()
strstr()	ucfirst()
stristr()	ucwords()
substr()	explode()
str_replace()	

Arrays

- Its collection of data
- Its used to store and organize data
- Same array can hold nos, char, strings and so on

Types of Array

- One dimensional array – Array with numeric index
- Associate array – Each ID key is associated with a value
- Multi dimensional array – An array containing another array

Array Declaration

- Use **array()** to declare an array

Arrays

One Dimensional Array Declaration

- Use **array()** to declare an array

Syntax

$array_name1=array();

$array_name2=array(“val1”,”val2”);

$array_name3[]=”val1”;

$array_name3[]=”val2”;

Eg:

$color_arr=array(“red”,”green”,”blue”,”yellow”);

Two Dimensional Array Declaration:

$array_name=array(key1=>val1,key2=>val2,key3=>val3);

Eg:

$StuInfo= array(“Raghu”=>1001,”Ram”=>1002,”Raja”=>1003);

Arrays

Array Function

count()	array_merge()
sizeof()	array_keys()
reset()	array_values()
array_push()	shuffle()
array_pop()	each()
array_unshift()	list()
array_shift()	

Advanced PHP

Date and Time

- PHP supports date and time related functions

Functions

- getdate()
- date()
- time()

Advanced PHP

getdate()

- It returns today's date, month, year, seconds, minutes, hours to array

Eg:-

$date_array=getdate();

date()

- Display the date and time as user's requirement

Syntax:

$str=date(format,timestamp);

Eg:

$str=date(“d/m/Y”);

Advanced PHP

Format – Day and Week

Format	Description	Returned Value
d	Day of the month with leading zeros	01, 02 to 31
j	Day of the month without leading zeros	1, 2 to 31
D	Textual representation of day in three char	Sun, Mon...
l	Full textual representation of day	Sunday, Monday to Saturday
N	ISO-8601 numeric representation of day	1-Monday, 2-Tuesday ..
S	Suffix for the day	St, nd, rd, th 1st, 2nd
w	Numeric representation of day	0-Sunday 1-Monday
z	Day of the year	0 to 365

Advanced PHP

Format – Month and Year

Format	Description	Returned Value
F	Full Textual representation of month	January to December
M	Short textual representation of month	Jan, Feb to Dec
m	Numeric representation of month with leading zeros	01, 02 to 12
n	Numeric representaion of month without leading zeros	1,2 to 12
t	Total no of days in the given month	28, 29, 30 and 31
L	Whether given year is leap year or not	1 – leap year 0 – Not a leap year
o	ISO-8601 Numeric representation of a year – 4 digit	1999, 2010
Y	Numeric representation of year – 4 digit	1999, 2010
y	Numeric representation of year – 2 digit	99, 10

Advanced PHP

Format – Time

Format	Description	Returned Value
a	Lowercase Ante Meridiem and Post Meridiem	`am or pm
A	Uppercase Ante Meridiem and Post Meridiem	AM or PM
B	Swatch internet time	000 to 999
g	12 hour time format without leading zero	1 to 12
G	24 hour time format without leading zero	0 to 23
h	12 hour time format with leading zero	01 to 12
H	24 hour time format with leading zero	00, 01 to 23
i	Minutes with leading zero	00 to 59
s	Seconds with leading zero	00 to 59
u	Microseconds	eg- 651241

Advanced PHP

Format	Description	Returned Value
e	Timezone identifier	Eg – GMT, UTC
I	Whether or not the date is in daylight saving time	1 – daylight saving time 0 - otherwise
O	Difference to GMT time in hours(no colon between hour and min)	Eg- +0530, -0230
P	Difference to GMT time in hours with colon between hour and min	Eg- +05:30, -02:30
T	Timezone abbreviation	IST, CST, PST, EST
Z	Timezone offset in seconds. The offset for timezone west of UTC is always negative and for those east of UTC is always positive.	-43200 to 50400
c	ISO 8601 date	2010-10-24T13:33:39+02:00
r	Formatted date	Sun, 24 Oct 2010 13:34:40+0200

Advanced PHP

PHP Include

- Insert the content of one php file into another php
- It can be done using **include()** and **require()**
- Both the functions are same but handles error differently

include()	**require()**
Generates a warning message if the file is not exist	Generates a fatal error if the file is not exist
Script will continue the execution	Script will stop the execution

Syntax

- include(“someFile.php”);
- require(“someFile.php”);

Advanced PHP

PHP File Handling

- Allows user to access a file
- It can be done using **fopen()** and **fclose()**

Syntax

- $fileName=fopen(filename,mode);

Eg:

```
<?php
$fileName=fopen(“someFile.txt”,”r”);
//Code
fclose($fileName);
<?
```

Advanced PHP

Mode

Mode	Description
r	Read Only
r+	Read/Write
w	Write Only
w+	Read/Write
a	Append. Opens and writes to the file or creates a new file if it does not exist
a+	Read/Append
x	Write Only, Creates a new file, Returns FALSE if file already exist
x+	Read/write, Creates a new file, Returns FALSE if file already exist.

Advanced PHP

PHP File Upload

- Allows user to upload a file from HTML file
- $_FILES is used to upload a file

Eg:

$_FILES[“file”][“name”] => Name of the uploaded file

$_FILES[“file”][“type”] => Type of the uploaded file

$_FILES[“file”][“size”] => Size of the uploaded file

$_FILES[“file”][“tmp_name”] => The name of the temporary copy of the file stored on the server

$_FILES[“file”][“error”] => Error code

Advanced PHP

Save Uploaded File

- Uploaded file will not be stored
- Copy of uploaded file is stored in temporary server

Eg:

```
<?php
if (file_exists("upload/" . $_FILES["file"]["name"]))
{
    echo $_FILES["file"]["name"] . " already exists. ";
}
else
{
    $tmpName=$_FILES[“file”][“tmp_name”];
    move_uploaded_file($tmpName,"upload/".$_FILES["file"]["name"]);
    echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
?>
```

Advanced PHP

Cookies

- HTTP Protocol is Stateless protocol
- Http never store user's info
- Identify user info
- Its a small file it includes user info at browser/client side
- Browser max can store 20 files
- Max size of an individual file is 4KB

Advanced PHP

Create a Cookie

- setcookie() - Creates cookie

Syntax

- setcookie(name,value,expire_time);

Eg:

- setcookie(“user”,”Raghu”,time()+3600);//Expire in one Hr

Retrieve a Cookie Value

- Variable $_COOKIE is used to retrieve a cookie value

Eg:

- $UserName=$_COOKIE[“user”];

Retrieve all Cookie Values

- print_r($_COOKIE);

Advanced PHP

Delete a Cookie

- Same setcookie() is used to delete a cookie

Syntax

- setcookie(name,Null_Value,time()-3600);

Eg:

- setcookie(“user”,””,time()-3600);

Eg:

```
<?php
setcookie("user","Raghu",time()+3600);
if(isset($_COOKIE["user"]))
echo "Welcome ".$_COOKIE["user"]."<br>";
else
echo "Welcome guest<br>";
?>
```

Advanced PHP

Drawbacks of Cookie

- Its not safety since its stored in client side
- Anyone can read Cookie file easily

Advanced PHP

Sessions

- Similar to cookies
- User info stored in Server side in /tmp
- Each client identified by Unique no → Session Id
- Session Id is stored in cookie
- User can store more no of variables
- It maintains user info across the web pages

Start Session

- session_start() - starts new session
- It generates session Id

Advanced PHP

Sessions Variable

- $_SESSION – Retrieves session variable

Syntax

$_SESSION['var_name']

Eg:

- $name=$_SESSION['user'];

Remove Session Data

- unset() - Removes session variable

Syntax

unset($_SESSION['var_name']);

Eg:

unset($_SESSION['user']);

Advanced PHP

End a Session

- session_destroy() - Destroys all session variable

Syntax

session_destroy();

```
<?php
session_start();
if(isset($_SESSION['visit']))
{
    $_SESSION['visit']++;
    echo "No of visit: ".$_SESSION['visit']."<br>";
}else
{
    echo "Session data is not available<br>";
    $_SESSION['visit']=0;
}?>
```

Thank you