Learning PHP

PHP started out as a small open source project that evolved as more and more people found out how useful it was.

Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

PHP is a recursive acronym for "PHP: Hypertext Preprocessor".

PHP is a server side scripting language that is embedded in HTML.

It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side.

The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

Install PHP

To install PHP, we will suggest you to install AMP (Apache, MySQL, PHP) software stack. Itis available for all operating systems. There are many AMP options available in the market that are given below:

- WAMP for Windows
- LAMP for Linux
- MAMP for Mac

PHP Syntax (save file .php)

```
<?php
```

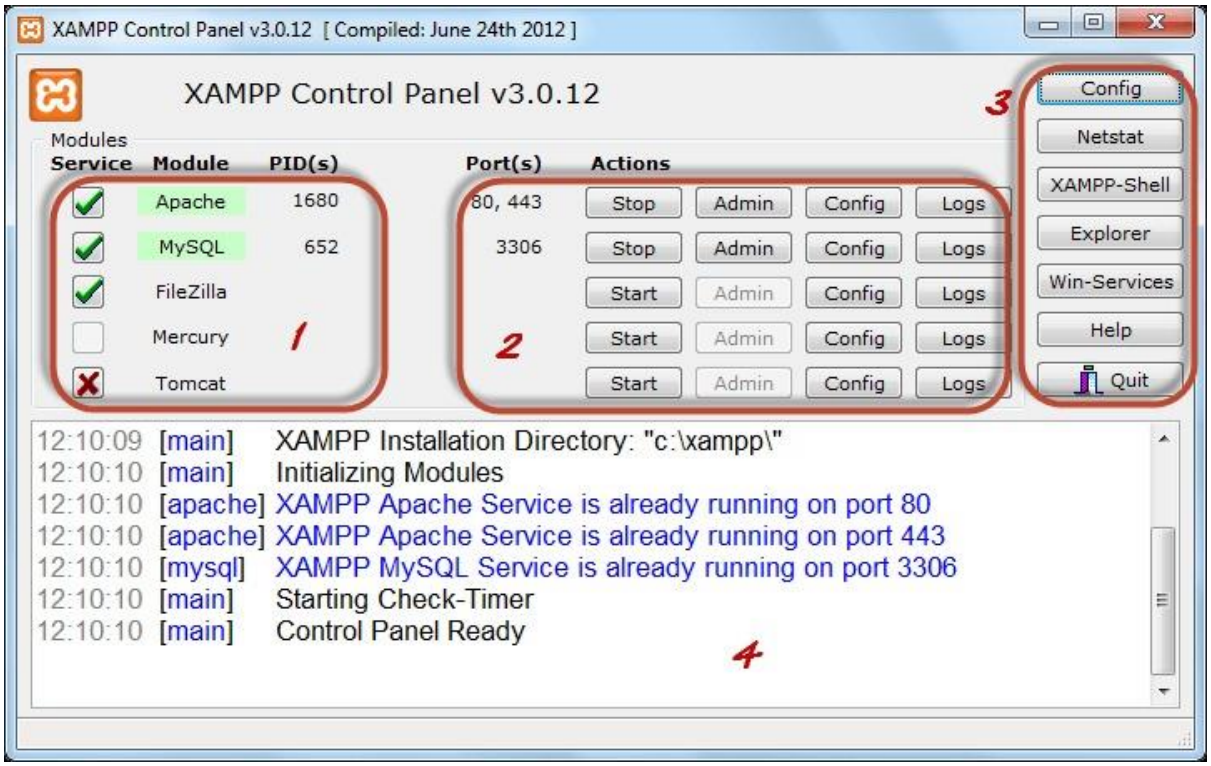
// some code here.....

?>

```
<?php
echo "Hello world";
?>
```

Basic XAMPP Web Server Configuration (If XAMPP installed)

This XAMPP Tutorial assumes that you have **installed XAMPP on drive C in Windows** The following is a list of the basic directories that you are supposed to be aware of.



PHP Data Types

A Data type is the classification of data into a category according to its attributes;

- Alphanumeric characters are classified as strings
- Whole numbers are classified integers
- Numbers with decimal points are classified as floating points.

- True or false values are classified as Boolean.

PHP is a loosely typed language; it does not have explicit defined data types. PHP determines the data types by analyzing the attributes of data supplied. PHP implicitly supports the following data types

- Integer – whole numbers e.g. -3, 0, 69. The maximum value of an integer is platform-dependent. On a 32 bit machine, it's usually around 2 billion. 64 bit machines usually have larger values. The constant PHP_INT_MAX is used to determine the maximum value.
- Floating point number – decimal numbers e.g. 3.14. they are also known as double or real numbers. The maximum value of a float is platform-dependent. Floating point numbers are larger than integers.
- Character string – e.g. Hello World
- Boolean – e.g. True or false.

PHP Variable

A variable is a name given to a memory location that stores data at runtime.
The scope of a variable determines its visibility.
A Php global variable is accessible to all the scripts in an application.
A local variable is only accessible to the script that it was defined in.
Think of a variable as a glass containing water. You can add water into the glass, drink all of it, refill it again etc.

```
<?php
$my_var = 1;
echo $my_var;
?>
```

```
<?php
$a = 1;
$b = 1.5;
$c = $a + $b;
$c = $a + (int) $b;
echo $c;
?>
```

PHP Constant

Define constant– A constant is a variable whose value cannot be changed at runtime. Suppose we are developing a program that uses the value of PI 3.14, we can use a constant to store its value.

PHP Operators

Arithmetic operators

Arithmetic operators are used to perform arithmetic operations on numeric data. The concatenate operator works on strings values too. PHP supports the following operators.

Operator	Name	Description	Example	Output
+	Addition	Summation of x and y	1 + 1;	2
–	Subtraction	Difference between x and y	1 – 1;	0
*	Multiplication	Multiplies x and y	3 * 7;	21
/	Division	Quotient of x and y	45 / 5;	9
%	PHP Modulus	Gives remainder of dividing x and y	10 % 3;	1
-n	Negation	Turns n into a negative number	-(-5);	5
x . y	Concatenation	Puts together x and y	“PHP” . ” ROCKS”;10 . 3;	PHP ROCKS103

Logical operators

When working with logical operators, any number greater than or less than zero (0) evaluates to true. Zero (0) evaluates to false.

Operator	Name	Description	Example	Output
X and y, x && y	And	Returns true if both x and y are equal	1 and 4;True&& False;	True or 1False or 0

X or y, x || y	Or	Returns true if either x or y is true	6 or 9;0 || 0;	True or 1False or 0
X xor y	Exclusive or, xor	Returns true if only x is true or only y is true	1 xor 1;1 xor 0;	False or 0True or 1
!x	Not	Returns true if x is false and false if x is true	!0;	True or 1

Comparison operators

Comparison operators are used to compare values and data types.

Operator	Name	Description	Example	Output
X == y	Equal	Compares x and y then returns true if they are equal	1 == "1";	True or 1
X === y	identical	Compares both values and data types.	1 === "1";	False or 0. Since 1 is integer and "1" is string
X != y, x <> y	PHP Not equal	Compares values of x and y. returns true if the values are not equal	2 != 1;	True or 1
X > y	Greater than	Compares values of x and y. returns true if x is greater than y	3 > 1;	True or 1
X < y	Less than	Compares values of x and y. returns true if x is less than y	2 < 1;	False or 0
X >= y	Greater than or equal	Compares values of x and y. returns true if x is greater than or equal to y	1 >=1	True or 1
X <= y	Less than or equal	Compares values of x and y. returns true if x is greater than or equal to y	8 <= 6	False or 0

Assignment Operators

Assignment operators are used to assign values to variables. They can also be used together with arithmetic operators.

Operator	Name	Description	Example	Output
x = ?	assignment	Assigns the value of x to ?	$x = 5;	5

x += ?	addition	Increments the value of x by ?	$x = 2;$x += 1;	3
X -= ?	subtraction	Subtracts ? from the value of x	$x = 3;$x -= 2;	1
X *=?	multiplication	Multiplies the value of x ? times	$x = 0;$x *=9;	0
X /=?	division	Quotient of x and ?	$x = 6;$x /=3;	2
X %=?	modulus	The reminder of dividing x by?	$x = 3;$x %= 2;	1
X .=?	concatenate	Puts together items	” $x = ‘Pretty’;$x .= ‘ Cool!’;”	Pretty Cool!

PHP Include & PHP Include_once

The “include” php statement is used to include other files into a PHP file.
It has two variations, include and include_once. Include_once is ignored by the PHP interpreter if the file to be included.
The include statement has the following syntax

```
<?php
include 'file_name';
?>
```

The include_once statement has the following syntax

```
<?php
include_once 'file_name';
?>
```

PHP Require & PHP require_once

The require statement has two variations, require and require_once.
The require/require_once statement is used to include file.
Require_once is ignored if the required file has already been added by any of the four include statements.
It has the following syntax

```
<?php
require 'file_name';
?>
```

```
<?php
require_once 'file_name';
?>
```

PHP include vs require

The difference between include / require

Include	Require
Issues a warning when an error occurs	Does not issue a warning
Execution of the script continues when an error occurs	Execution of the script stops when an error occurs.

PHP Comments

- Comments help us to understand the code
- Comments are explanations that we include in our source code. These comments are for human understanding.
- Single line comments start with double forward slashes // and they end in the same line.
- //
- Multiple line comments start with a forward slash followed by the asterisk /* and end with the asterisk followed by the forward slash */.

```
/*this is a multiple-line
*comment
```

- /*example

What is a PHP Array?

A PHP array is a variable that stores more than one piece of related data in a single variable.
Think of an array as a box of chocolates with slots inside.

PHP Associative Array

Associative array differ from numeric array in the sense that associative arrays use descriptive names for id keys.
Below is the syntax for creating associative array in php.

```
<?php
$variable_name['key_name'] = value;

$variable_name = array('keyname' => value);
?>
```

Numeric Arrays

Numeric arrays use number as access keys.
An access key is a reference to a memory slot in an array variable.

```
<?php

$movie[0] = 'Shaolin Monk';
$movie[1] = 'Drunken Master';
$movie[2] = 'American Ninja';
$movie[3] = 'Once upon a time in China';
$movie[4] = 'Replacement Killers';

?>
```

PHP Multi-dimensional arrays

These are arrays that contain other nested arrays.
The advantage of multidimensional arrays is that they allow us to group related data together.

```
<?php
$movies =array(
"comedy" => array("Pink Panther", "John English", "See no evil
hear no evil"),
```

```
"action" => array("Die Hard", "Expendables"),
"epic" => array("The Lord of the rings"),
"Romance" => array("Romeo and Juliet")
);
print_r($movies);
?>
```

PHP Array Functions

Count function

The count function is used to count the number of elements that an php array contains. The code below shows the implementation.

```
<?php
$lecturers = array("Mr. Jones", "Mr. Banda", "Mrs. Smith");
echo count($lecturers);
?>
```

is_array function

The is_array function is used to determine if a variable is an array or not. Let's now look at an example that implements the is_array functions.

```
<?php
$lecturers = array("Mr. Jones", "Mr. Banda", "Mrs. Smith");
echo is_array($lecturers);
?>
```

Sort

This function is used to sort arrays by the values.
If the values are alphanumeric, it sorts them in alphabetical order.
If the values are numeric, it sorts them in ascending order.
It removes the existing access keys and add new numeric keys.
The output of this function is a numeric array

```
<?php
$persons = array("Mary" => "Female", "John" => "Male",
"Mirriam" => "Female");
```

```
sort($persons);

print_r($persons);
?>
```

ksort

This function is used to sort the array using the key. The following example illustrates its usage.

```
<?php
$persons = array("Mary" => "Female", "John" => "Male",
"Mirriam" => "Female");

ksort($persons);

print_r($persons);
?>
```

asort

This function is used to sort the array using the values. The following example illustrates its usage.

```
<?php

$persons = array("Mary" => "Female", "John" => "Male",
"Mirriam" => "Female");

asort($persons);

print_r($persons);

?>
```

---PROGRAMS---

How to Add Two Numbers using PHP?

```
?php

if(isset($_POST['submit']))
```

```
{
$num1=$_POST['num1'];
$num2=$_POST['num2'];
$result=$num1+$num2;
echo "Sum of " .$num1. " and ".$num2. " is " .$result;
}
?>
<!DOCTYPE html>
<html>
<head>
<title>Add two number</title>
</head>
<body>
<table>
<form name="add" method="post">
<tr>
<td>Number 1 :</td>
<td><input type="text" name="num1" required></td>
</tr>
<tr>
<td>Number 2 :</td>
<td><input type="text" name="num2" required></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Add" name="submit" /></td>
</tr>
</form>
</table>
```

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

How find Even odd Numbers in PHP?

```
<?php
if(isset($_POST['submit']))
{
$num=$_POST['num'];
if($num%2==0){
  echo "$num  is a even number ";
}
else
{
echo "$num is a odd number";
}
}
?>
<!DOCTYPE html>
<html>
<head>
<title>Even odd  Numbers</title>
</head>
<body>
<table>
<form name="evenodd" method="post">
<tr>
<td>Enter Number :</td>
<td><input type="text" name="num" required></td>
```

```
</tr>
<tr>
<td></td>
<td><input type="submit" value="Check" name="submit"
/></td>
</tr>
</form>
</table>
</body>
</html>
```

How to Print Table of any Number in PHP?

```
?php
if(isset($_POST['submit']))
{
$num=$_POST['num'];
define('NUM',$num);
for($i=1 ; $i<=10 ; $i++)
{
echo $i*NUM;
echo '<br>';
}
}
?>
<!DOCTYPE html>
<html>
<head>
<title>Table of a Number</title>
</head>
```

```
<body>
<table>
<form name="table" method="post">
<tr>
<td>Enter Number :</td>
<td><input type="text" name="num" required></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="submit" name="submit"
/></td>
</tr>
</form>
</table>
</body>
</html>
```

How to Reverse Number using PHP?

```
<?php
if(isset($_POST['submit']))
{
$n=$_POST['num'];
function reverse_number($number)
{
$snum = (string) $number;
$revstr = strrev($snum);
$reverse = (int) $revstr;
return $reverse;
```

```
}
echo reverse_number($n);
}
?>
<!DOCTYPE html>
<html>
<head>
<title>Reverse a number </title>
</head>
<body>
<table>
<form name="reverse" method="post">
<tr>
<td>Enter Number :</td>
<td><input type="text" name="num" required></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Reverse" name="submit"
/></td>
</tr>
</form>
</table>
</body>
</html>
```