PHP

Audio Book Companion

Troy Dimes

All rights Reserved. No part of this publication or the information in it may be quoted from or reproduced in any form by means such as printing, scanning, photocopying or otherwise without prior written permission of the copyright holder.

Disclaimer and Terms of Use: Effort has been made to ensure that the information in this book is accurate and complete, however, the author and the publisher do not warrant the accuracy of the information, text and graphics contained within the book due to the rapidly changing nature of science, research, known and unknown facts and internet. The Author and the publisher do not hold any responsibility for errors, omissions or contrary interpretation of the subject matter herein. This book is presented solely for motivational and informational purposes only.

Contents

Chapter 1: Introduction & Installation 1

Chapter 2: PHP Basic Syntax 6

Chapter 3: Understanding Variables & String Functions 10

Chapter 4: PHP Operators 16

Chapter 5: Iteration and Control Statements 23

Chapter 6: Understanding PHP Arrays 28

Chapter 7: Functions in PHP 34

Chapter 8: Object Oriented PHP (Part 1) 40

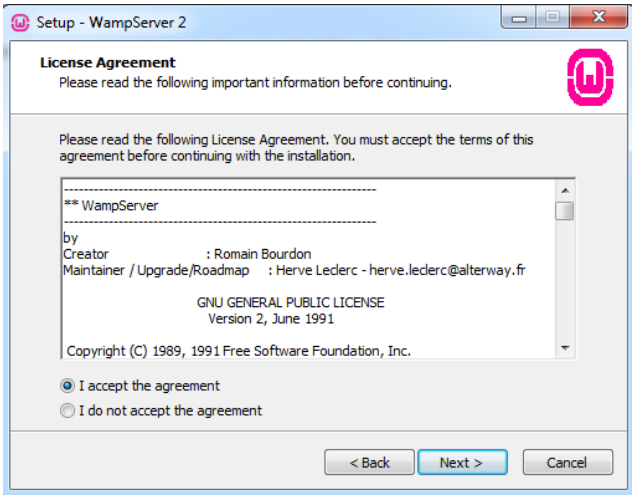
Chapter 9: Object Oriented PHP (Part 2) 45

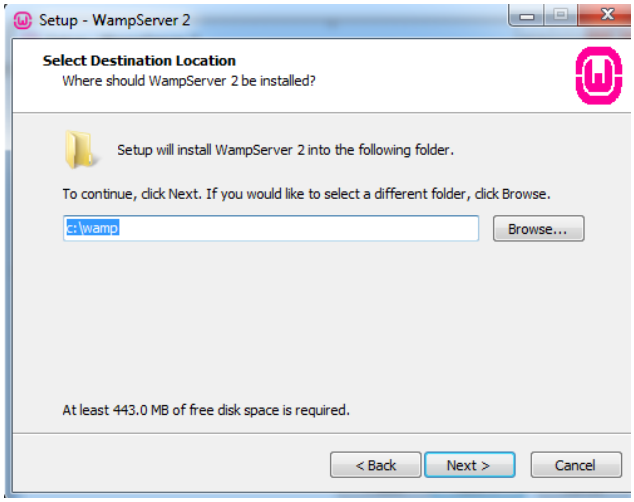
Chapter 10: Understanding GET and POST Methods 52

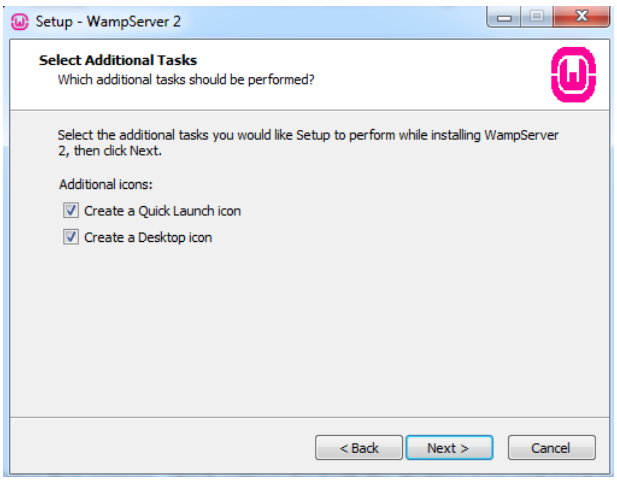
Other Books by the Author 57

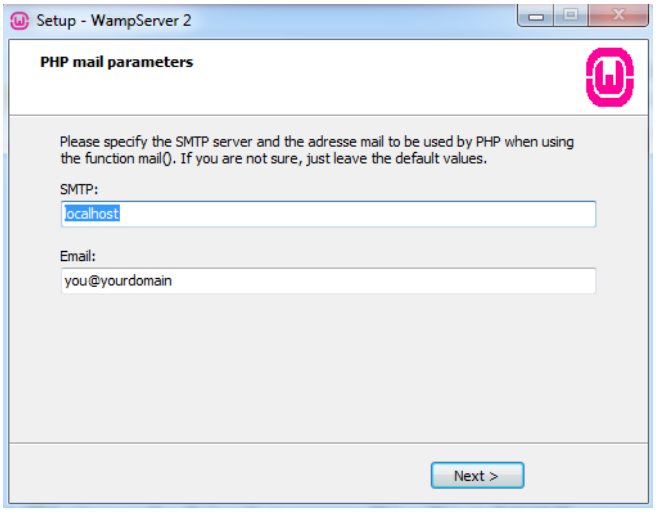
Additional Resources 59

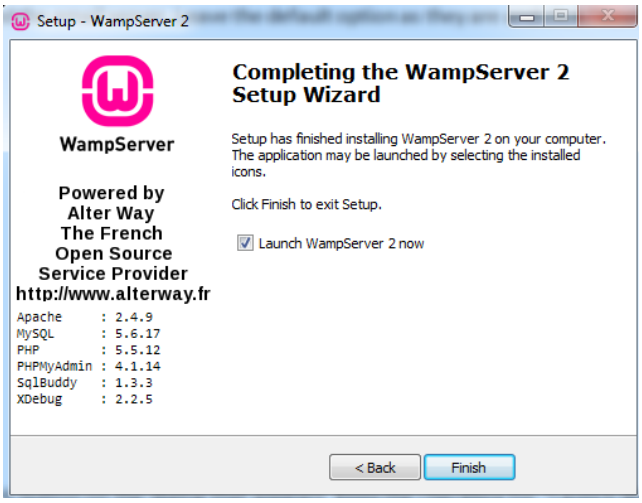
Chapter 1: Introduction & Installation

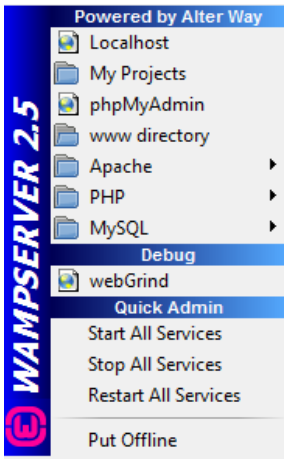

Setup - WampServer 2
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
** WampServer
by
Creator : Romain Bourdon
Maintainer / Upgrade/Roadmap : Herve Leclerc - herve.leclerc@alterway.fr
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
I accept the agreement
I do not accept the agreement
< Back
Next >
Cancel

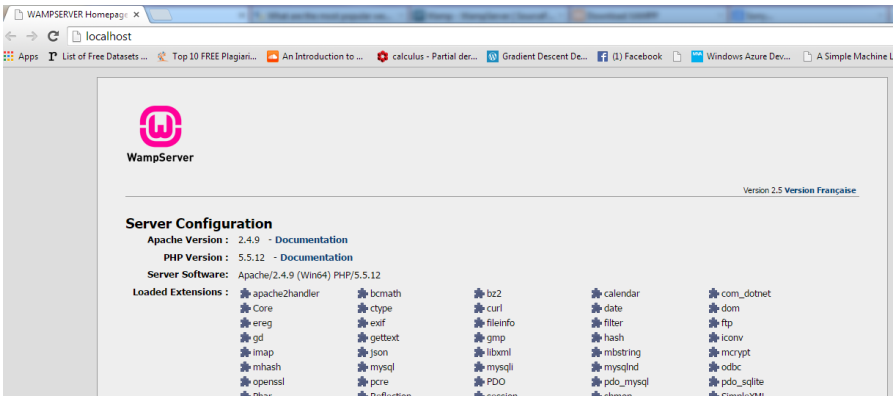

Setup - WampServer 2
Select Destination Location
Where should WampServer 2 be installed?
Setup will install WampServer 2 into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
c:\wamp
Browse...
At least 443.0 MB of free disk space is required.
< Back
Next >
Cancel

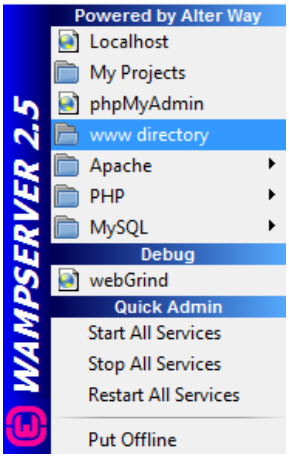

Setup - WampServer 2
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing WampServer 2, then click Next.
Additional icons:
Create a Quick Launch icon
Create a Desktop icon
< Back
Next >
Cancel


Setup - WampServer 2
PHP mail parameters
Please specify the SMTP server and the adresse mail to be used by PHP when using the function mail(). If you are not sure, just leave the default values.
SMTP:
localhost
Email:
you@yourdomain
Next >


Setup - WampServer 2
WampServer
Powered by
Alter Way
The French
Open Source
Service Provider
http://www.alterway.fr
Apache : 2.4.9
MySQL : 5.6.17
PHP : 5.5.12
PHPMyAdmin : 4.1.14
SqlBuddy : 1.3.3
XDebug : 2.2.5
Completing the WampServer 2 Setup Wizard
Setup has finished installing WampServer 2 on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch WampServer 2 now
< Back
Finish


Powered by Alter Way
Localhost
My Projects
phpMyAdmin
www directory
Apache
PHP
MySQL
Debug
webGrind
Quick Admin
Start All Services
Stop All Services
Restart All Services
Put Offline
WAMPSERVER 2.5


WAMPSERVER Homepage
localhost
WampServer
Version 2.5 Version Française
Server Configuration
Apache Version : 2.4.9 - Documentation
PHP Version : 5.5.12 - Documentation
Server Software: Apache/2.4.9 (Win64) PHP/5.5.12
Loaded Extensions :
apache2handler
bcmath
bz2
calendar
com_dotnet
Core
ctype
curl
date
dom
ereg
exif
fileinfo
filter
ftp
gd
gettext
gmp
hash
iconv
imap
json
libxml
mbstring
mcrypt
mhash
mysql
mysqli
mysqlnd
odbc
openssl
pcre
PDO
pdo_mysql
pdo_sqlite


Powered by Alter Way
Localhost
My Projects
phpMyAdmin
www directory
Apache
PHP
MySQL
Debug
webGrind
Quick Admin
Start All Services
Stop All Services
Restart All Services
Put Offline
WAMPSERVER 2.5

```
favicon
index.php
testmysql.php
```

Chapter 2: PHP Basic Syntax

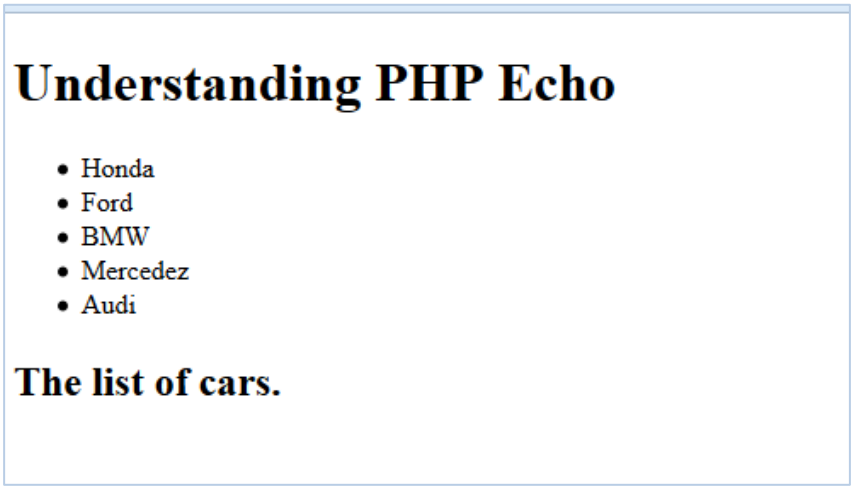
Example 1:

```
 <!DOCTYPE html>
<html>
<head>
<title>Example1</title>
</head>

<body>
<h1>This will go as it is. </h1>

<?php
echo "This is parsed by PHP";
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Example1</title>
</head>

<body>
<h1>This will go as it is. </h1>

This is parsed by PHP</body>
</html>
```

Example 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Example2</title>
</head>

<body>
<h1>Understanding PHP Echo </h1>

<?php
echo "<ul>
<li> Honda</li>
<li> Ford</li>
<li> BMW</li>
<li> Mercedez</li>
<li> Audi</li>

</ul>";

echo "<h2>The list of cars.</h2>";
?>
</body>

</html>
```



Example 3:

```
<!DOCTYPE html>
<html>
<head>
<title>Example3</title>
</head>

<body>
<h1>Understanding PHP Comments </h1>

<?php

/*
The following
echo statement
prints list of cars.
*/
echo "<ul>
<li>Honda</li>
<li>Ford</li>
<li>BMW</li>
<li>Mercedez</li>
<li>Audi</li>

</ul>";

// The following echo statement prints a message.
echo "<h2>The list of cars.</h2>";
```

```
?>
</body>

</html>
```

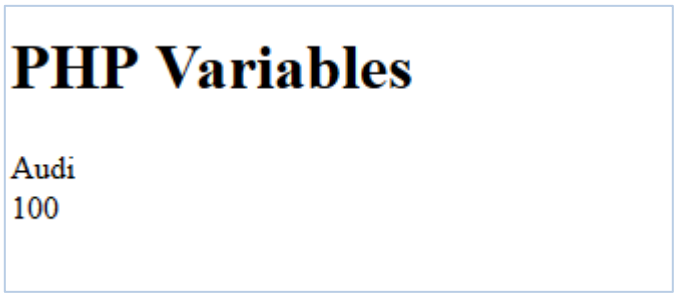
Solution:

```
<!DOCTYPE html>
<html>
<head>
<title>Exercise 2</title>
</head>

<body>
<h1>Exercise 2 Solution: </h1>

<body>
<?php
/* The following echo statement
 creates first button and text box
 */
echo '<button> Button 1 </button> <input type="text"/> <br>';

/* The following echo statement
 creates second button and text box
 */
echo '<button> Button2 </button> <input type="text"/> <br>';
?>
</body>

</html>
```

Chapter 3: Understanding Variables & String Functions

Example 1:

```
<!DOCTYPE html>
<html>
<head>
<title>Example 1</title>
</head>

<body>
<h1>PHP Variables</h1>

<body>
<?php

$car_name = 'Audi';
$car_price = 100;

echo $car_name;
echo '<br>';
echo $car_price;
?>
</body>

</html>
```




```
<!DOCTYPE html>
<html>
<head>
<title>Example 2</title>
</head>

<body>
<h1>Adding two variables</h1>

<body>
<?php

$a = 5;
$b = 10;
$result = $a + $b;

echo "Result:";
echo $result;

?>
</body>

</html>
```

```php
<?php

$a = 5;
$b = 10;
$result = $a + $b;

echo "Result:".$result;

?>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Example 3</title>
</head>

<body>
<h1>Getting length of a string</h1>

<body>

<?php
$car_name = "Audi";
echo  strlen($car_name);
$car_name = "Toyota";
echo "<br>";
echo  strlen($car_name);
?>

</body>

</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>String Functions</title>
</head>

<body>
<h1>Getting words in a string</h1>

<body>

<?php
$content = "Hello how are you?";
echo  str_word_count($content);
$content = "I am fine";
echo "<br>";
echo  str_word_count($content);
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>String Functions</title>
</head>

<body>
<h1>Reversing a string</h1>

<body>

<?php
$content = "Hello how are you?";
echo  strrev($content);
$content = "I am fine";
echo "<br>";
echo  strrev($content);
?>

</body>

</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>String Functions</title>
</head>

<body>
<h1>Replacing content within a string</h1>

<body>

<?php
$content = "Hello how are you?";
echo $content;
echo "<br>";
echo  str_replace("how", "where",$content);
?>
</body>
</html>
```

Solution:

```
<!DOCTYPE html>
<html>
<head>
<title>String Functions</title>
</head>

<body>
<h1>Exercise 3 Solution</h1>

<body>

<?php
$weather_info = "It is cold today";

echo "<br>";
echo  strlen($weather_info);
echo "<br>";
echo str_word_count($weather_info);
echo "<br>";
echo strrev($weather_info);

?>

</body>

</html>
```

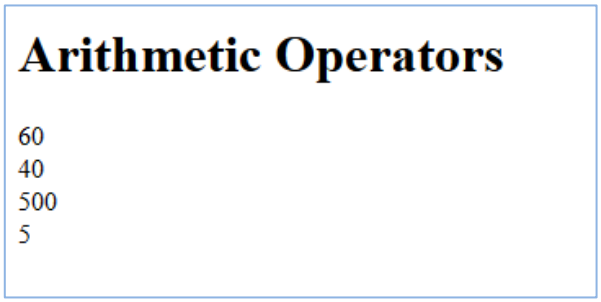
Chapter 4: PHP Operators

Operator	Name	Example	Output
+	Addition	$a+ $b	Sum of $a and $b
-	Subtraction	$a- $b	Difference of $a and $b
*	Multiplication	$a* $b	Product of $a and $b
/	Division	$a/ $b	Quotient of $a and $b
%	Modulus	$a% $b	Remainder of $a divided by $b
**	Exponentiation	$a** $b	Result of raising $ a to the $b'th power

Example 1:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 4</title>
</head>

<body>
```

```
<h1>Arithmetic Operators</h1>

<body>

<?php
$a = 50;
$b = 10;

echo $a + $b;
echo "<br>";
echo $a - $b;
echo "<br>";
echo $a * $b;
echo "<br>";
echo $a / $b;

?>

</body>

</html>
```



Operator	Name	Example	Output
==	Equal	$a == $b	Returns true if $a is equal to $b
===	Identical	$a === $b	Returns true if $a is equal to $b, and they are of the same type
!=	Not equal	$a != $b	Returns true if $a is not equal to $b
<>	Not equal	$a <> $b	Returns true if $a is not equal to $b
!==	Not identical	$a !== $b	Returns true if $a is not equal to $b, or they are not of the same type
>	Greater than	$a > $b	Returns true if $a is greater than $b
<	Less than	$a < $b	Returns true if $a is less than $b
>=	Greater than or equal to	$a >= $b	Returns true if $a is greater than or equal to $b
<=	Less than or equal to	$a <= $b	Returns true if $a is less than or equal to $b

Assignment	Similar to	What it does
a = b	a = b	The left operand is set to the value of the operand on the right
a += b	a = a + b	Addition
a -= b	a = a - b	Subtraction
a *= b	a = a * b	Multiplication
a /= b	a = a / b	Division
a %= b	a = a % b	Modulus

Example 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 4</title>
</head>

<body>
<h1>Assignment Operators</h1>

<body>

<?php
$a = 50;
$b = 10;

$a += $b;

echo $a;
echo"<br>";
$a = 50;

$a -= $b;

echo $a;
echo"<br>";
$a = 50;

$a *= $b;

echo $a;
echo"<br>";
$a = 50;

$a /= $b;

echo $a;
echo"<br>";
$a = 50;

$a %= $b;

echo $a;

?>
```

```
</body>

</html>
```

Operator	Name	Example	Output
and	And	$a and $b	True if both $a and $b are true
or	Or	$a or $b	True if either $a or $b is true
xor	Xor	$a xor $b	True if either $a or $b is true, but not both
&&	And	$a && $b	True if both $a and $b are true
||	Or	$a || $b	True if either $a or $b is true
!	Not	!$a	True if $a is not true

Operator	Name	What it does
++$a	Pre-increment	Increments $a by one, then returns $a
$a++	Post-increment	Returns $a, then increments $a by one
--$a	Pre-decrement	Decrements $a by one, then returns $a
$a--	Post-decrement	Returns $a, then decrements $a by one

Example 3:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 4</title>
</head>

<body>
<h1>Increment & Decrement Operators</h1>

<body>

<?php
$a = 45;
echo ++$a;
echo "<br>";
echo $a++;
echo "<br>";
echo --$a;
echo "<br>";
echo $a--;
?>

</body>

</html>
```

Solution

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 4</title>
</head>

<body>
<h1>Exercise 4 Solution.</h1>

<body>

<?php
$number1 = 36;
$number2 = 6;

echo $number1 + $number2;
echo "<br>";
echo $number1 - $number2;
echo "<br>";
echo $number1 * $number2;
echo "<br>";
echo $number1 / $number2;

?>


</body>

</html>
```

Chapter 5: Iteration and Control Statements

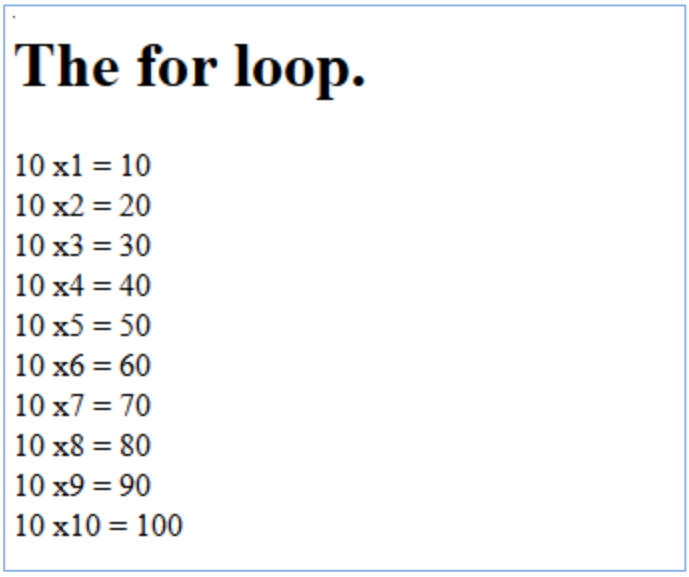
Example 1:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 5</title>
</head>

<body>
<h1>The for loop.</h1>

<body>

<?php

for($k=1; $k<=10; $k++)
{
        $output = 10*$k;
        echo "<li> 10 x".$k." = ".$output."</li>";
}
?>

</body>

</html>
```

The above piece of code will print the table of 10 on the output webpage. Have a look at the “for” loop in the above code:

```
for($k=1; $k<=10; $k++)
```

```
{
        $output = 10*$k;
        echo "<li> 10 x".$k." = ".$output."</li>";
}
```



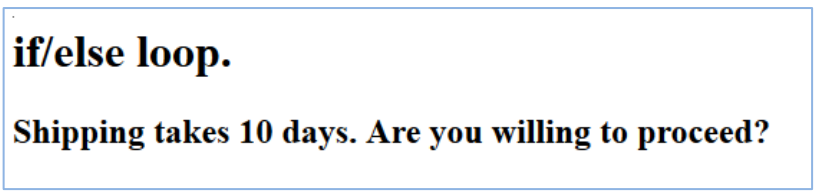
Example 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 5</title>
</head>

<body>
<h1>The while loop.</h1>

<?php
$k=1;
while($k<=10)
{
        $output = 10*$k;
        echo "<li> 10 x".$k." = ".$output."</li>";
        $k++;
}
?>
```

```
</body>

</html>
```

Example 3:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 5</title>
</head>

<body>
<h1>if/else loop.</h1>

<body>

<?php

$country = "uvw";

if($country =="abc" || $country == "xyz")
{
        echo "<h2>Shipping Services Not Available</h2>";
}
else if ($country=="uvw")
{
        echo '<h2>Shipping takes 10 days. Are you willing to proceed?</h2>';
}
else
{
        echo "<h2> Your order has been placed.</h2>";
}
?>

</body>

</html>
```



Example 4:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 5</title>
</head>

<body>
<h1>Switch Statements.</h1>

<body>

<?php

$country = "uvw";

switch ($country)
{
case "abc":
echo '<h2>Shipping Services Not Available</h2>';
break;

case "xyz":
echo '<h2>Shipping Services Not Available</h2>';
break;

case "uvw":
echo '<h2>Shipping takes 10 days. Are you willing to proceed?</h2>';
break;

default:
echo '<h2> Your order has been placed.</h2>';
break;
}

?>

</body>

</html>
```

Switch Statements.

Shipping takes 10 days. Are you willing to proceed?

Solution

```
<!DOCTYPE html>
<html>
<head>
<title>Exercise 5</title>
</head>

<body>
<h1>Exercise 5 Solution.</h1>

<body>

<?php

$output = 1;
$input=7;
$k=1;
while( $k<=$input)
{
        $output = $k * $output;
        $k++;
}
echo $output;

?>

</body>
</html>
```

Chapter 6: Understanding PHP Arrays

Example 1:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 6</title>
</head>

<body>
<h1>Arrays in PHP</h1>

<body>

<?php

$countries[0]= "USA";
$countries[1]= "RUSSIA";
$countries[2]= "CANADA";
$countries[3]= "CHINA";
$countries[4]= "ENGLAND";

$countries[]= "FRANCE";
$countries[]= "ITALY";
$countries[]= "GERMANY";
$countries[]= "SPAIN";
$countries[]= "PORTUGAL";

print_r($countries);

?>
```

```
</body>

</html>
```

Arrays in PHP

Array ([0] => USA [1] => RUSSIA [2] => CANADA [3] => CHINA [4] => ENGLAND [5] => FRANCE [6] => ITALY [7] => GERMANY [8] => SPAIN [9] => PORTUGAL)

Example 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 6</title>
</head>

<body>
<h1>Using Array Keyword and Foreach</h1>

<body>

<?php

$countries = array('USA', 'CANADA', 'CHINA', 'FRANCE', 'ENGLAND');

foreach($countries as $country)
{
        echo $country."<br>";
}

?>

</body>

</html>
```

Example 3:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 6</title>
</head>

<body>
<h1>Array Traversal</h1>

<body>

<?php

$countries[0]= "USA";
$countries[1]= "RUSSIA";
$countries[2]= "CANADA";
$countries[3]= "CHINA";
$countries[4]= "ENGLAND";

$countries[]= "FRANCE";
$countries[]= "ITALY";
$countries[]= "GERMANY";
$countries[]= "SPAIN";
$countries[]= "PORTUGAL";


echo '<ul>';
for($x=0;$x<10;$x++)
{
        echo '<li>'.$countries[$x].'</li>';
}
echo '</ul>';

?>


</body>

</html>
```

Example 4:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 6</title>
</head>

<body>
<h1>Associative Array</h1>

<body>

<?php

$countries['America1']= "USA";
$countries['Europe1']= "RUSSIA";
$countries['America2']= "CANADA";
$countries['Asia1']= "CHINA";
$countries['Europe2']= "ENGLAND";

echo $countries['Asia1'].'<br>';
echo $countries['Europe1'].'<br>';

?>

</body>
</html>
```

Associative Array

CHINA
RUSSIA

Solution:

```
<!DOCTYPE html>
<html>
<head>
<title>Exercise 6</title>
</head>

<body>
<h1>Exercise 6 Solution</h1>

<body>

<?php

$integers = array(48,15,45,65,26,78,19,35,64,76);

$largest=0;
$smallest = 0;

foreach($integers as $integer)
{
        if($integer > $largest)
        {
                $largest = $integer;
        }

        if($smallest == 0)
        {
                $smallest = $integer;
        }
        else if($integer < $smallest)
        {
                $smallest = $integer;
        }
}

echo $largest.'<br>';
echo $smallest;

?>


</body>

</html>
```

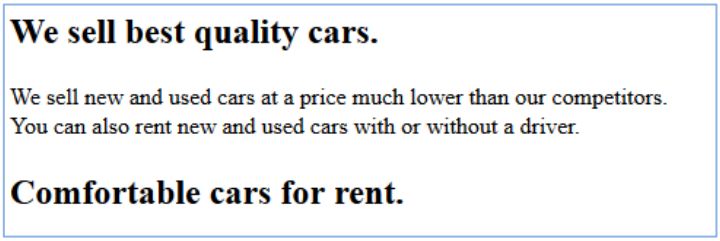
Chapter 7: Functions in PHP

Example 1:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 7</title>
</head>

<body>

<?php

function displayTagLine()
{
        echo "<h1>A Place for Top Cars </h1>";
}

displayTagLine();
 echo "<p> We sell new and used cars at a price much lower than our competitors.
<br>
 You can also rent new and used cars with or without a driver.</p>";

displayTagLine();

?>

</body>

</html>
```

Example 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 7</title>
</head>

<body>

<?php

function displayTagLine($message)
{
        echo '<h2>'.$message.'</h2>';
}

displayTagLine('We sell best quality cars.');
 echo "<p> We sell new and used cars at a price much lower than our competitors.
<br>
 You can also rent new and used cars with or without a driver.</p>";

displayTagLine('Comfortable cars for rent.');

?>

</body>

</html>
```

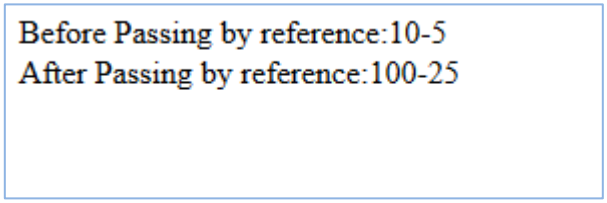


Example 3:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 7</title>
</head>

<body>

<?php

function sumOfSquares($num1, $num2)
{
        return ($num1* $num1) + ($num2 * $num2);
}
echo sumOfSquares(2,3);

?>

</body>

</html>
```

Example 4:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 7</title>
</head>

<body>

<?php

$num1=10;
$num2 = 5;

echo 'Before Passing by reference:'.$num1.'-'.$num2.'<br>';
SquareByRef($num1, $num2);
echo 'After Passing by reference:'.$num1.'-'.$num2;

function SquareByRef(&$a, &$b)
{
        $a= $a * $a;
        $b = $b * $b;
}

?>

</body>

</html>
```



Solution

```
<!DOCTYPE html>
<html>
<head>
<title>Exercise 7</title>
</head>

<body>

<?php

$integers = array(48,15,45,65,26,78,19,35,64,76);
$integers2 = array(41,25,65,47,89,12,35,4,76,98);
$integers3 = array(12,36,47,58,96,32,56,41,59,74);


processArray($integers);
processArray($integers2);
processArray($integers3);

function processArray($array)
{
$largest=0;
$smallest = 0;

foreach($array as $item)
{
        if($item > $largest)
        {
                $largest = $item;
        }

        if($smallest == 0)
        {
                $smallest = $item;
        }
        else if($item < $smallest)
        {
                $smallest = $item;
        }
}

echo '<br><br>';
echo $largest.'<br>';
echo $smallest.'<br>';
```

```
}

?>

</body>

</html>
```

Chapter 8: Object Oriented PHP (Part 1)

Example 1:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 8</title>
</head>

<body>

<?php

class  User{
    /* Member variables */
    var $name;
        var $email;
    var $password;

    /* Member functions */
    function signUp($uname, $upassword, $uemail){

           $this->name = $uname;
           $this->password = $upassword;
           $this->email = $uemail;
    }

        function getName()
        {
        return $this->name;
        }
}
```

```
?>


</body>

</html>
```

Example 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 8</title>
</head>

<body>

<?php

class  User{
    /* Member variables */
    var $name;
        var $email;
    var $password;

    /* Member functions */
    function signUp($uname, $upassword, $uemail){

           $this->name = $uname;
           $this->password = $upassword;
           $this->email = $uemail;
    }

        function getName()
        {
        return $this->name;
        }
}

$user1 = new User;
$user2 = new User;

$user1->signUp("Jack", "12345", "abc@xyz.com");

$user2->signUp("Mark", "879789", "abcd@xyz1.com");
```

```
echo $user1->getName().'<br>';
echo $user1->email.'<br>';
echo $user2->getName().'<br>';
?>


</body>

</html>
```

Jack
abc@xyz.com
Mark

Example 3:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 8</title>
</head>

<body>

<?php

class  User{
    /* Member variables */
    var $name;
        var $email;
    var $password;

    /* Member functions */
    function __construct($uname, $upassword, $uemail){

           $this->name = $uname;
           $this->password = $upassword;
           $this->email = $uemail;
    }
```

```
        function getName()
        {
        return $this->name;
        }
}

$user1 = new User("Jack", "12345", "abc@xyz.com");
$user2 = new User("Mark", "879789", "abcd@xyz1.com");

echo $user1->getName().'<br>';
echo $user1->email.'<br>';
echo $user2->getName().'<br>';
?>


</body>

</html>
```

```
$user1 = new User("Jack", "12345", "abc@xyz.com");
```

Solution:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 8</title>
</head>

<body>

<?php

class  Item{
    /* Member variables */
    var $name;
        var $price;
    var $category;

    /* Member functions */
    function __construct($iname, $iprice, $icategory){

            $this->name = $iname;
            $this->price = $iprice;
            $this->category = $icategory;
    }

        function getCategory()
        {
        return $this->category;
        }
}

$item1 = new Item("TV", "400", "Electronics");
$item2 = new Item("Table", "200", "Furniture");

echo $item1->getCategory().'<br>';
echo $item2->getCategory().'<br>';
?>


</body>

</html>
```

Chapter 9: Object Oriented PHP (Part 2)

Example 1:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 9</title>
</head>

<body>

<?php

class Item
{
        var $name;
        function setName($name)
        {
                $this->name = $name;
        }

        function getName()
        {
                return $this->name;
        }
}

class Laptop extends Item
{
        var $price;
        function setPrice($price)
        {
```

```
                $this->price = $price;
        }

        function getPrice()
        {
                return $this->price;
        }
}

$laptop1 = new Laptop;

$laptop1->setName("Sonny Laptop");
$laptop1->setPrice(450);

echo $laptop1->getName().'<br>';
echo $laptop1->getPrice();

?>

</body>

</html>
```

Example 2:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 9</title>
</head>

<body>

<?php

class Item
{
        var $name;
```

```
        function setName($name)
        {
                $this->name = $name;
        }

        function getName()
        {
                return $this->name;
        }
}

class Laptop extends Item
{
        var $price;
        function setPrice($price)
        {
                $this->price = $price;
        }

        function getPrice()
        {
                return $this->price;
        }

        function getName()
        {
                return "this is a child function";
        }
}

$laptop1 = new Laptop;

$laptop1->setName("Sonny Laptop");
$laptop1->setPrice(450);

echo $laptop1->getName().'<br>';
echo $laptop1->getPrice();

?>

</body>

</html>
```

```
this is a child function
450
```

Example 3:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 9</title>
</head>

<body>

<?php

class Item
{
        var $name;
        function setName($name)
        {
                $this->name = $name;
        }

        function getName()
        {
                return $this->name;
        }
}

class Laptop extends Item
{
        var $price;
        function __construct($name, $price)
        {
                Item::setName($name);
                $this->price = $price;
        }

        function getPrice()
        {
                return $this->price;
```

```
        }

}

$laptop1 = new Laptop("Sonny", 120);


echo $laptop1->getName().'<br>';
echo $laptop1->getPrice();

?>

</body>

</html>
```

Example 4:

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 9</title>
</head>

<body>

<?php

class Item
{
        var $name;
        public static $price;
        function setName($name)
        {
                $this->name = $name;
        }

        function getName()
        {
                return $this->name;
        }
}

$item1 = new Item;
$item1->setName("Cell Phone");
Item::$price = 500;
```

```
echo $item1->getName().'<br>';
echo Item::$price;

?>

</body>

</html>
```

Solution

```
<!DOCTYPE html>
<html>
<head>
<title>Exercise 9</title>
</head>

<body>

<?php

class Shape
{
        var $name;
        function setProperties($name)
        {
                $this->name = $name;
        }

        function getName()
        {
                return $this->name;
        }
}

class Square extends Shape
{
        var $width, $height;

        function __construct($name, $width, $height)
        {
                Shape::setProperties($name);
                $this->width =$width;
                $this->height = $height;
```

```
        }
        function getWidth()
        {
                return $this->width;
        }
        function getHeight()
        {
                return $this->height;
        }
}

$square = new Square("Square Shape", 40, 30);
echo $square->getName().'<br>';
echo $square->getHeight().'<br>';
echo $square->getWidth();

?>

</body>

</html>
```

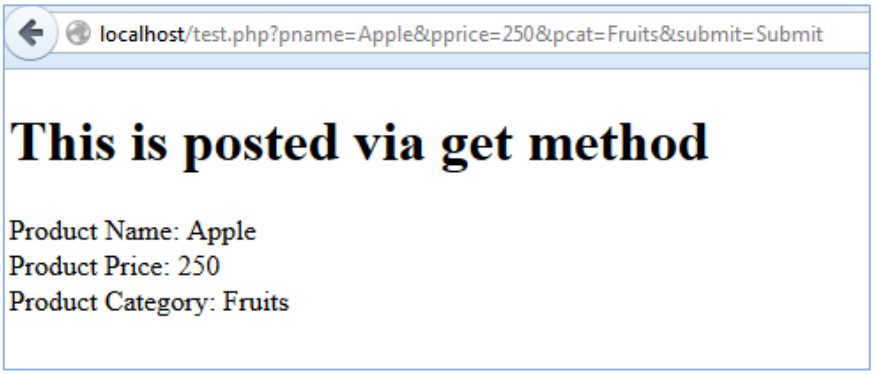
Chapter 10: Understanding GET and POST Methods

Example 1:

Create an HTML page with following content.

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 10</title>
</head>

<body>

<form action="http://localhost/test.php" method="get">
<input type="text" name="pname" placeholder="Product Name"></input><br/>
<input type="text" name="pprice" placeholder="Product Price"></input><br/>
<input type="text" name="pcat" placeholder="Product Category"></input><br/>
<input type="submit" name="submit" value="Submit"></input>
</form>


</body>

</html>
```

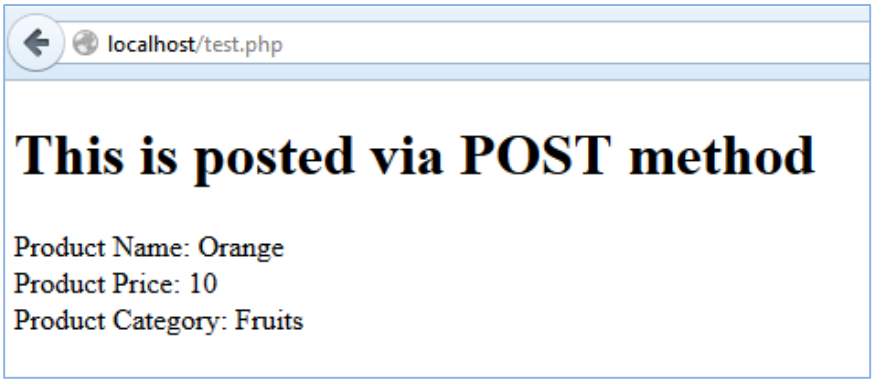
```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 10</title>
</head>

<body>

<h1>This is posted via get method</h1>
<?php
if( $_GET["pname"] || $_GET["pprice"] || $_GET["pcat"])
{
echo "Product Name: ". $_GET['pname']. "<br />";
echo "Product Price: ". $_GET["pprice"]. "<br />";
echo "Product Category: ". $_GET["pcat"];
}
?>

</body>

</html>
```



Example 2:

```
<form action="http://localhost/test.php" method="post">
<input type="text" name="pname" placeholder="Product Name"></input><br/>
<input type="text" name="pprice" placeholder="Product Price"></input><br/>
<input type="text" name="pcat" placeholder="Product Category"></input><br/>
<input type="submit" name="submit" value="Submit"></input>
</form>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Chapter 10</title>
</head>

<body>

<h1>This is posted via POST method</h1>
<?php
if( $_POST["pname"] || $_POST["pprice"] || $_POST["pcat"])
{
echo "Product Name: ". $_POST['pname']. "<br />";
echo "Product Price: ". $_POST["pprice"]. "<br />";
echo "Product Category: ". $_POST["pcat"];
}
?>
</body>

</html>
```



Solution

Form Page

```
<!DOCTYPE html>
<html>
<head>
<title>Exercise 10</title>
</head>

<body>

<form action="http://localhost/salary.php" method="post">
<input type="text" name="name" placeholder="Name"></input><br/>
<input type="text" name="salary" placeholder="Salary"></input><br/>
<input type="submit" name="submit" value="Submit"></input>
</form>


</body>

</html>
```

salary.php

```
<!DOCTYPE html>
<html>
<head>
<title>Exercise 10</title>
</head>

<body>

<?php
if( $_POST["name"] || $_POST["salary"])
{
echo "Name: ". $_POST['name']. "<br />";
echo "Salary: ". $_POST['salary']. "<br />";

$bonus = $_POST["salary"]/10;
echo "Bonus on Salary: ". $bonus;
}
?>
</body>
```

```
</html>
```

Other Books by the Author

C# Programming for Beginners
http://www.linuxtrainingacademy.com/c-sharp

C# is a simple and general-purpose object-oriented programming language. Combine this with its versatility and huge standard library it's easy to see why it's such a popular and well-respected programming language.

When you learn how to program in C# you will be able to develop web based applications or graphical desktop applications. One of the best things about C# is that it's easy to learn… especially with this book.

Java Programming
http://www.linuxtrainingacademy.com/java-programming

Java is one of the most widely used and powerful computer programming languages in existence today. Once you learn how to program in Java you can create software applications that run on servers, desktop computers, tablets, phones, Blu-ray players, and more.

Also, if you want to ensure your software behaves the same regardless of which operation system it runs on, then Java's "write once, run anywhere" philosophy is for you. Java was design to be platform independent allowing you to create applications that run on a variety of operating systems including Windows, Mac, Solaris, and Linux.

JavaScript: A Guide to Learning the JavaScript Programming Language
http://www.linuxtrainingacademy.com/javascript

JavaScript is a dynamic computer programming language that is commonly used in web browsers to control the behavior of web pages and interact with users. It allows for asynchronous communication and can update parts of a web page or even replace the entire content of a web page. You'll see JavaScript being used to display date and time information, perform animations on a web site, validate form input, suggest results as a user types into a search box, and more.

Scrum Essentials: Agile Software Development and Agile Project Management for Project Managers, Scrum Masters, Product Owners, and Stakeholders
http://www.linuxtrainingacademy.com/scrum-book

You have a limited amount of time to create software, especially when you're given a deadline, self-imposed or not. You'll want to make sure that the software you build is at least decent but more importantly, on time. How do you balance quality with time? This book dives into these very important topics and more.

Additional Resources

Learn PHP Programming from Scratch Course
http://www.linuxtrainingacademy.com/php-from-scratch

This is a comprehensive PHP course. You will learn everything from the basics to more advanced PHP programming techniques using real world examples and sample projects.