Chris Carrier WD 220

9/26/13 Chapter 4

#4-1

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Jumble Maker</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

</head>

<body>

<form action = "process\_JumbleMaker.php" method="post">

Word 1: <input type="text" name="Word1" /><br />

Word 2: <input type="text" name="Word2" /><br />

Word 3: <input type="text" name="Word3" /><br />

Word 4: <input type="text" name="Word4" /><br />

<input type="reset" value="Clear Form" />&nbsp;

&nbsp;<input type="submit" name="Submit" value="Send Form" />

</form>

<?php

?>

</body>

</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Jumble Maker</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

</head>

<body>

<?php

function displayError($fieldName, $errorMsg){

global $errorCount;

echo "Error for \"$fieldName\": $errorMsg<br />\n";

++$errorCount;

}

function validateWord($data, $fieldName){

global $errorCount;

if(empty($data)){

displayError($fieldName, "This is required");

$retval = "";

}

else{ // Only clean up the input if it isn't empty

$retval = trim($data);

$retval = stripslashes($retval);

if((strlen($retval) < 4) || (strlen($retval) > 7)){

displayError($fieldName, "Words must be at least four and at most seven letters long");

}

if(preg\_match("/^[a-z]+$/i",$retval) == 0){

displayError(fieldName, "Words must be only letters");

}

}

$retval = strtoupper($retval);

$retval = str\_shuffle($retval);

return($retval);

}

$errorCount = 0;

$words = array();

$words[] = validateWord($\_POST['Word1'], "Word 1");

$words[] = validateWord($\_POST['Word2'], "Word 2");

$words[] = validateWord($\_POST['Word3'], "Word 3");

$words[] = validateWord($\_POST['Word4'], "Word 4");

if($errorCount > 0){

echo "Please use the \"Back\" button to re-enter the data.<br />\n";

}

else{

$wordnum = 0;

foreach($words as $word)

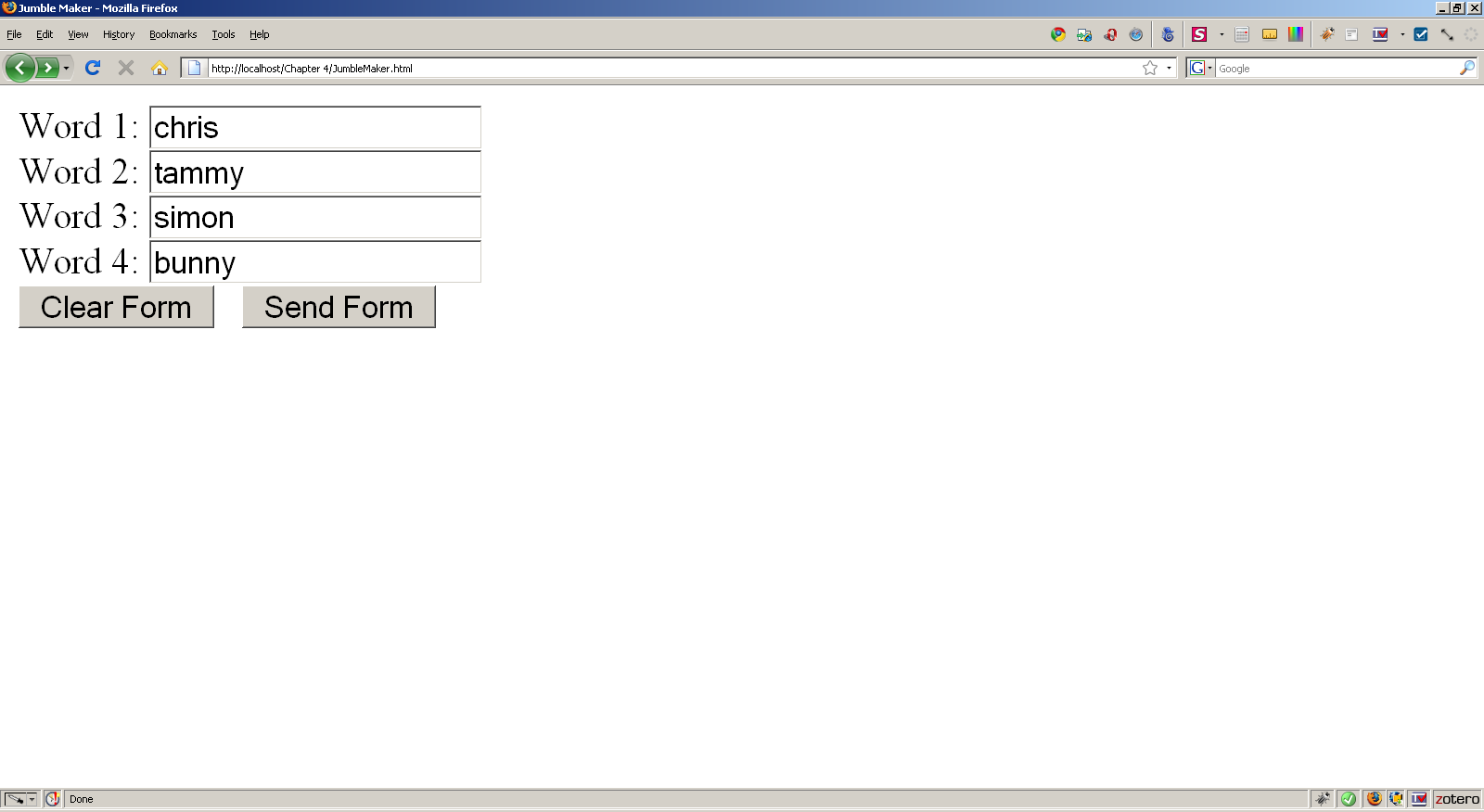
echo "Word ".++$wordnum.": $word<br />\n";

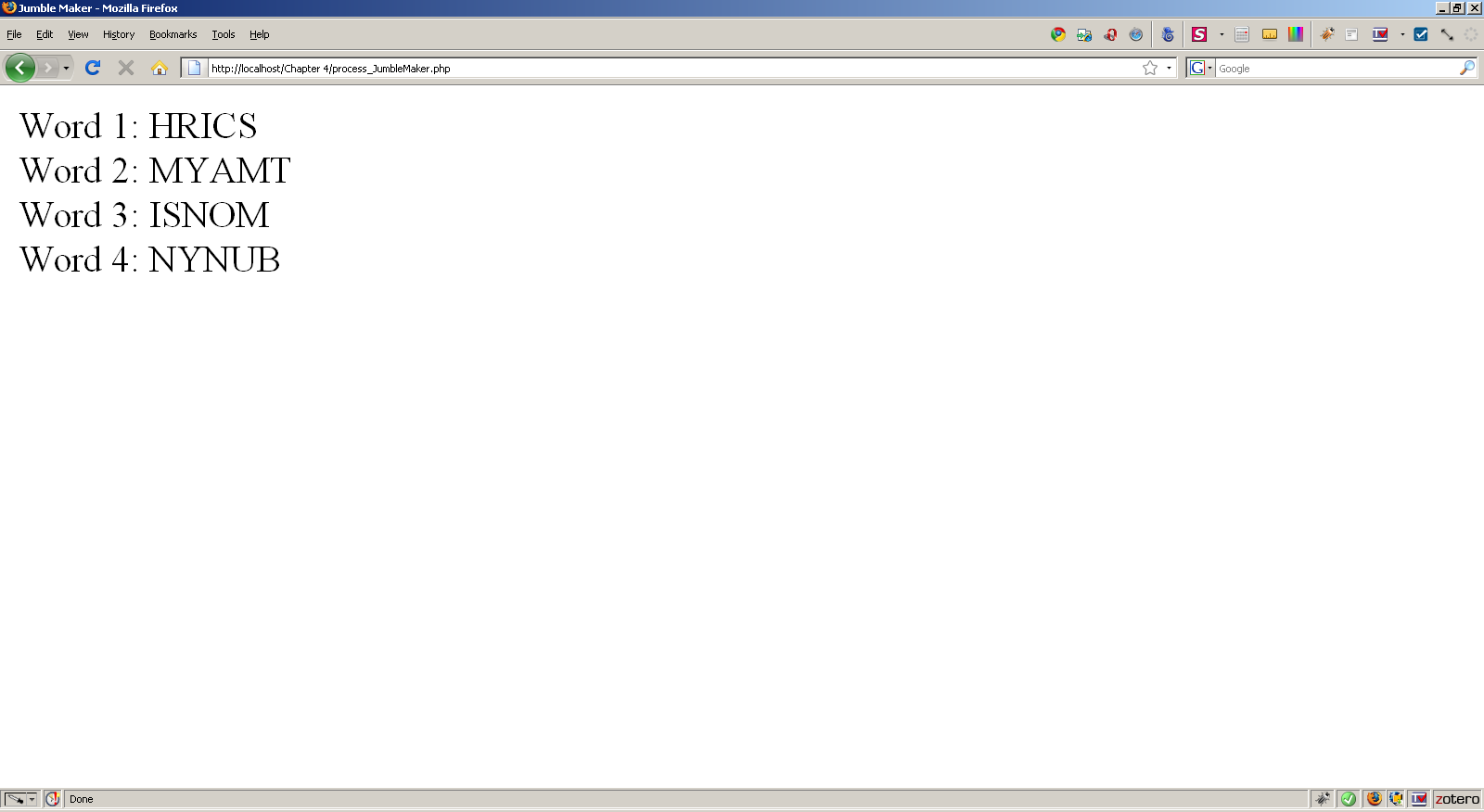
}

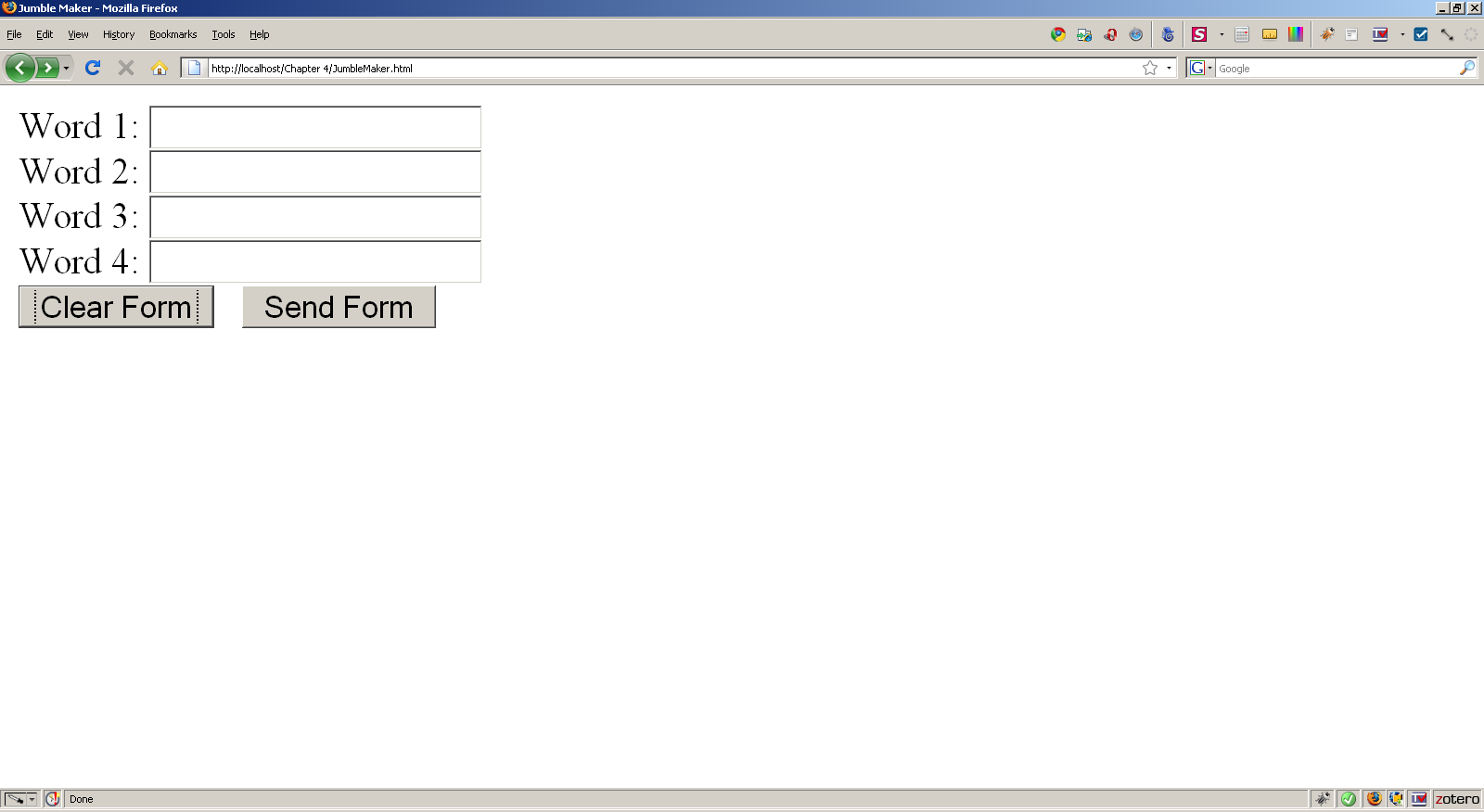
?>

</body>

</html>







# 4-3

Inc\_requestDump.php

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Include Request Dump</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

</head>

<body>

<?php

function displayError($fieldName, $errorMsg){

global $errorCount;

echo "Error for \"$fieldName\": $errorMsg<br />\n";

++$errorCount;

}

function validateWord($data, $fieldName){

global $errorCount;

if(empty($data)){

displayError($fieldName, "This is required");

$retval = "";

}

else{ // Only clean up the input if it isn't empty

$retval = trim($data);

$retval = stripslashes($retval);

$retval = htmlentities($retval);

if((strlen($retval) < 2) || (strlen($retval) > 20)){

displayError($fieldName, "Words must be at least four and at most twenty letters long");

}

if(preg\_match("/^[a-z]+$/i",$retval) == 0){

displayError(fieldName, "Words must be only letters");

}

}

$retval = strtoupper($retval);

return($retval);

}

$errorCount = 0;

$firstNames = array();

$lastNamses = array();

$fNames[] = validateWord($\_REQUEST['fName'], "First Name");

$lNames[] = validateWord($\_REQUEST['lName'], "Last Name");

$cities[] = validateWord($\_REQUEST['city'], "City");

$states[] = validateWord($\_REQUEST['state'], "State");

if($errorCount > 0){

echo "Please use the \"Back\" button to re-enter the data.<br />\n";

}

else{ // Output

$wordnum = 0;

foreach($fNames as $fName) // First Name

echo ++$wordnum.".) First Name: $fName<br />\n";

foreach($lNames as $lName) // Last Name

echo ++$wordnum.".) Last Name: $lName<br />\n";

foreach($cities as $city) // City

echo ++$wordnum.".) City: $city<br />\n";

foreach($states as $state) // State

echo ++$wordnum.".) State: $state<br />\n";

}

?>

</body>

</html>

RequestDump.php

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Request Dump</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

</head>

<body>

<h2 style="text-align:center">Request Dump</h2>

<form name="RequestDump" action="inc\_requestDump.php" method="post">

First Name: <input type="text" name="fName" /><br />

Last Name: <input type="text" name="lName" /><br />

City: <input type="text" name="city" /><br />

State: <input type="text" name="state" /><br />

<input type="reset" value="Clear Form" />&nbsp;

&nbsp;<input type="submit" name="Submit" value="Send Form" /><br />

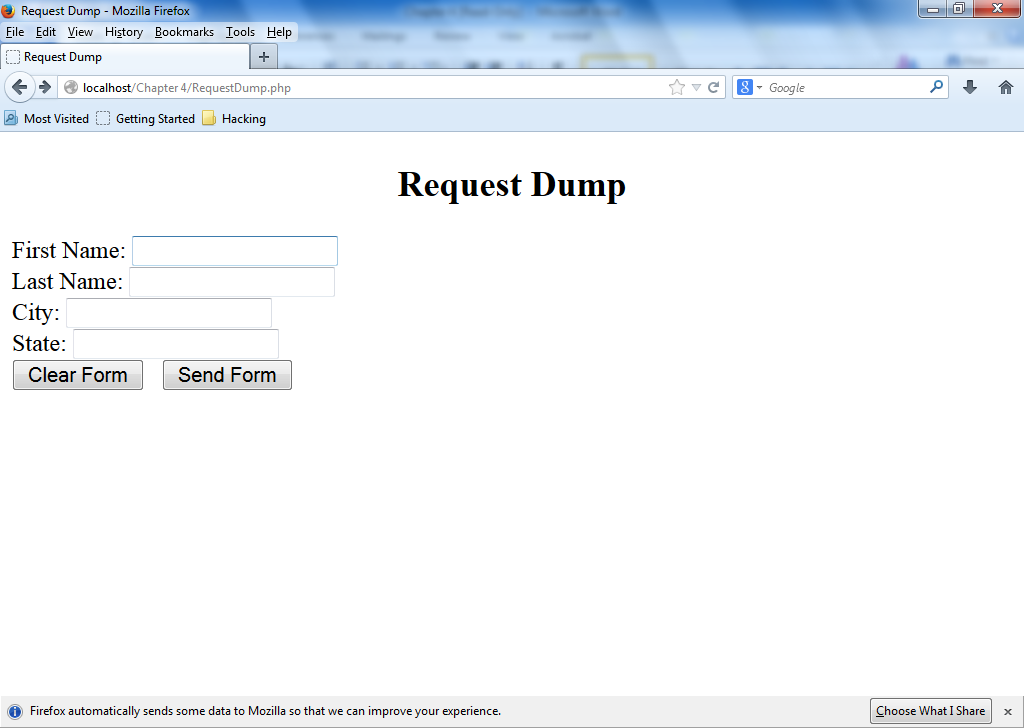
</form>

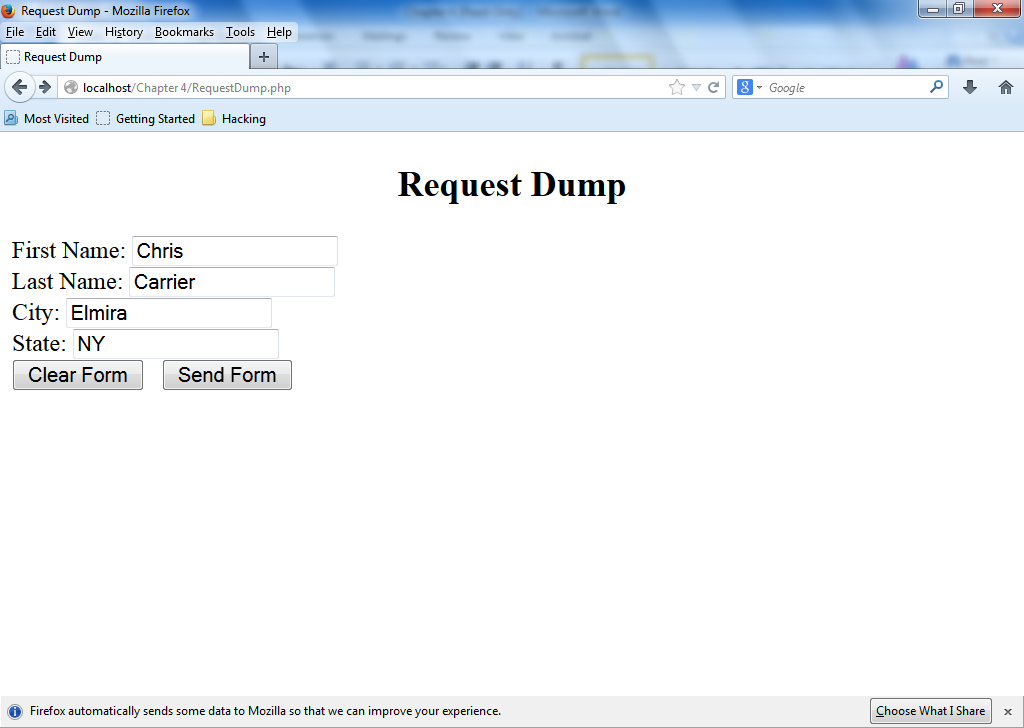
<?php

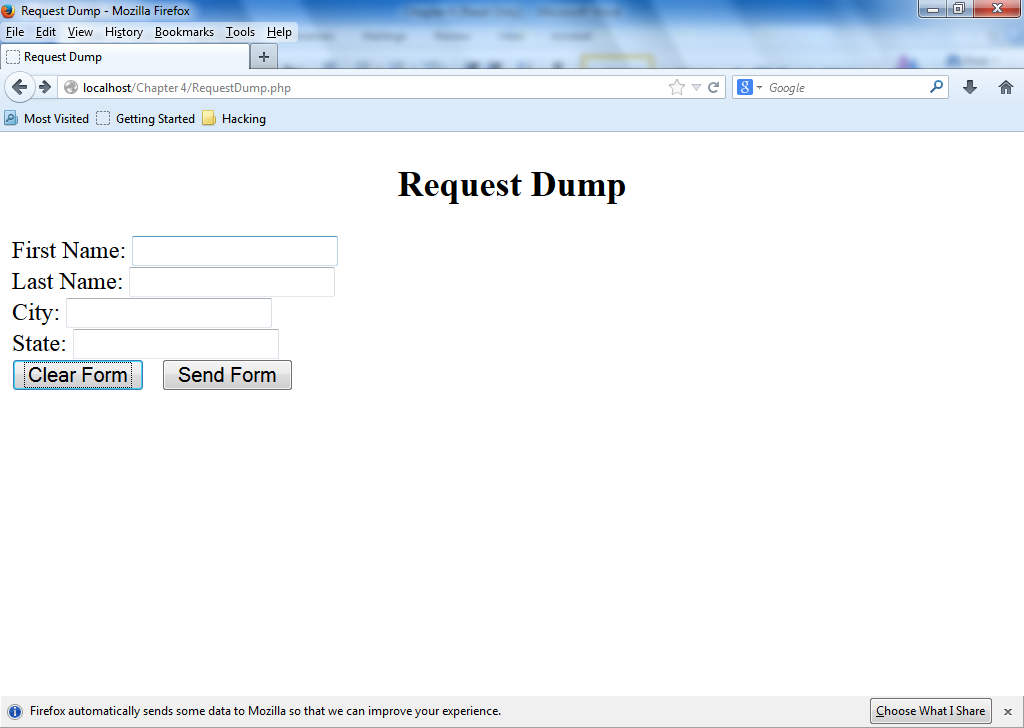
?>

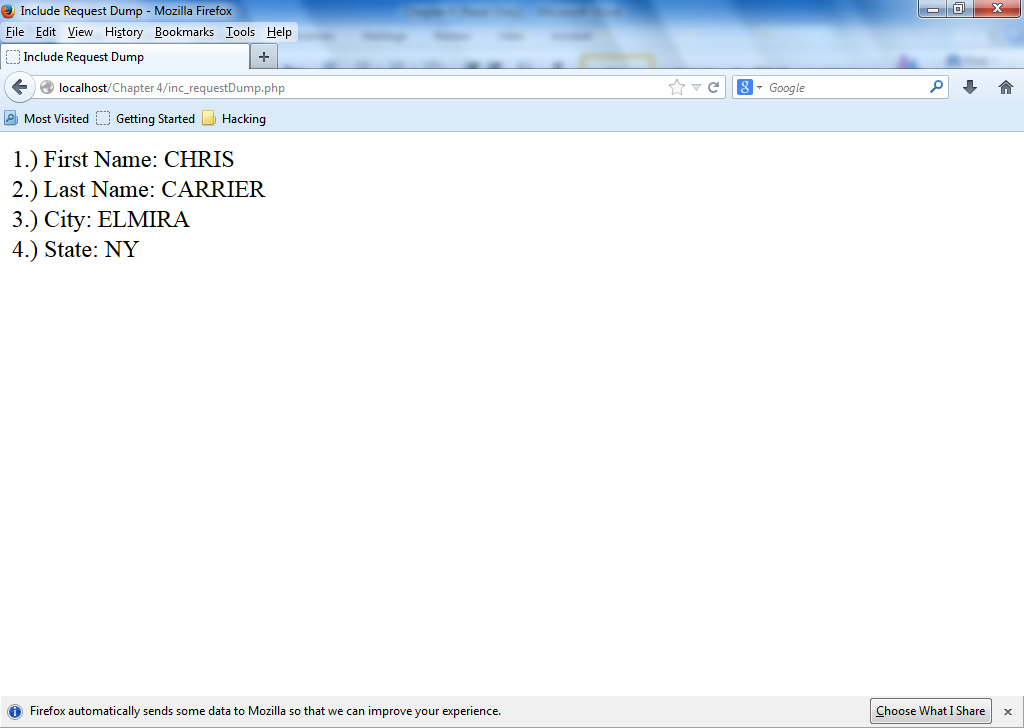
</body>

</html>









I couldn’t figure out how to make them columns. I have a feeling it’s probably right in front of me but oh well.

Exercise 4-4

paycheck.php

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Pay Check PHP</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

</head>

<body>

<?php

function displayError($fieldName, $errorMsg){

global $errorCount;

echo "Error for \"$fieldName\": $errorMsg<br />\n";

++$errorCount;

}

function validateWord($data, $fieldName){

global $errorCount;

if(empty($data)){

displayError($fieldName, "This is required");

$retvalue = "";

}

else{ // Only clean up the input if it isn't empty

$retvalue = trim($data);

$retvalue = stripslashes($retvalue);

if((strlen($retvalue) < 2) || (strlen($retvalue) > 10)){

displayError($fieldName, " Entries must be at least two and at most ten numbers long");

}

if(preg\_match("/^[0-9]+$/i", $retvalue) == 1){

displayError($fieldName, " Entries must be numbers only");

}

if(is\_numeric($retvalue)){

$retvalue = number\_format($retvalue, 2);

}

}

$retvalue = strtoupper($retvalue);

return($retvalue);

}

$errorCount = 0;

$overTime = 0;

$regularTime = 0;

$hoursworkeds = array();

$hourlywages = array();

$hoursworkeds[] = validateWord($\_POST['HoursWorked'], "HoursWorked");

$hourlywages[] = validateWord($\_POST['HourlyWage'], "HourlyWage");

if($errorCount > 0){

echo "Please use the \"Back\" button to re-enter the data.<br />\n";

}

else{

$num = 0;

foreach($hoursworkeds as $hoursworked)

echo ++$num.".) Hours Worked: $hoursworked<br />\n";

foreach($hourlywages as $hourlywage)

echo ++$num.".) Hourly Wage: $$hourlywage<br />\n";

echo "\n";

//Calculation

$hoursworkeds = $hoursworked;

$hourlywages = $hourlywage;

if($hoursworked > 40){

$overTime = (($hoursworked - 40) \* ($hourlywage \* 1.5));

$overTimeTotal = (($hourlywage \* 40) + $overTime);

$overTimeTotal = number\_format($overTimeTotal, 2);

printf("<p>Your paycheck for the week is: $%s.</p>\n", $overTimeTotal);

}

else{

$regularTime = $hoursworked \* $hourlywage;

$regularTime = number\_format($regularTime, 2);

printf("<p>Your paycheck for the week is: $%s.</p>\n", $regularTime);

}

}

?>

</body>

</html>

paycheck.html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Pay Check HTML</title>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

</head>

<body>

<form action="Paycheck.php" method="post">

Hours Worked: <input type="text" name="HoursWorked" required /><br />

Hourly Wage: $<input type="text" name="HourlyWage" required /><br />

<input type="reset" value="Clear Form" />&nbsp;

&nbsp;<input type="submit" name="Submit" value="Send Form" />

</form>

<?php

?>

</body>

</html>

