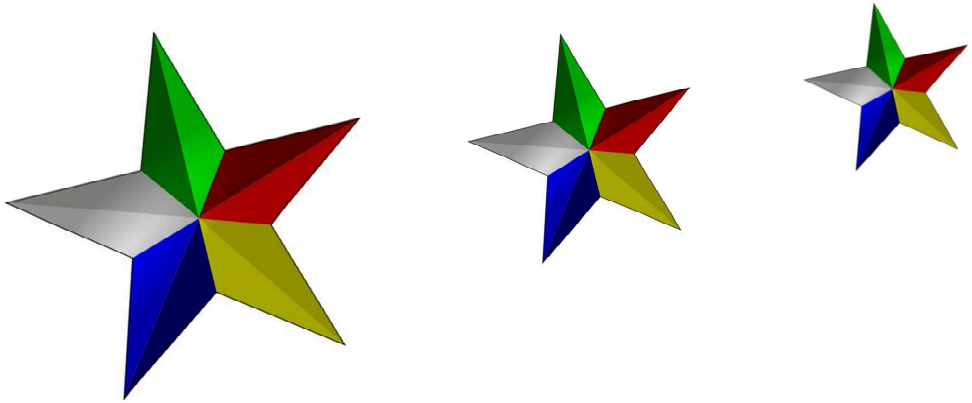
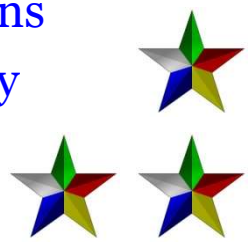


PHP Programming

Dr. M. Anand
Assistant Professor (Sr. G)
Dept. of Networking and Communications
SRM Institute of Science and Technology



PHP

- **PHP: Hypertext Preprocessor**
- most popular server-side scripting languages for creating dynamic web pages
- PHP was created by Rasmus Lerdorf

Simple PHP Program

- PHP, code is inserted between the scripting delimiters **<?php** and **?>**

```php
<?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.1: first.php -->
<!-- Simple PHP program. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
<?php
   $name = "Harvey"; // declaration and initialization
?><!-- end PHP script -->
   <head>
      <title>Using PHP document</title>
   </head>
   <body style = "font-size: 2em">
      <p>
         <strong>
            <!-- print variable name's value -->
            Welcome to PHP, <?php print( "$name" ); ?>!
         </strong>
      </p>
   </body>
</html>
```

Variables

- All variables are preceded by a **$**
- Variables are created the first time they are encountered by the PHP interpreter
- PHP statements terminate with a **semicolon (;)**

Variables

- *Variable names in PHP are case sensitive*
  - *PHP script will create a new variable for any name it doesn't recognize as a previously used variable.*

Eg.:

```
$max=10;
$Max=100;
echo $Max;
```

Comments

- **single-line comment**, which begins with two forward slashes (//). Text to the right of the slashes is ignored by the interpreter.
- Single-line comments can also begin with the pound sign (#)
- **Multiline comments** begin with delimiter **/*** and end with delimiter ***/**

```
<?php print( "$name" ); ?>
```

- The actual value of $name is printed, not the string "$name".
- When a variable is encountered inside a double-quoted ("") string, PHP **interpolates** the variable
  - **In other words, PHP inserts the** variable's value where the variable name appears in the string.
  - Thus, variable $name is replaced by Harvey for printing purposes.

Data Types

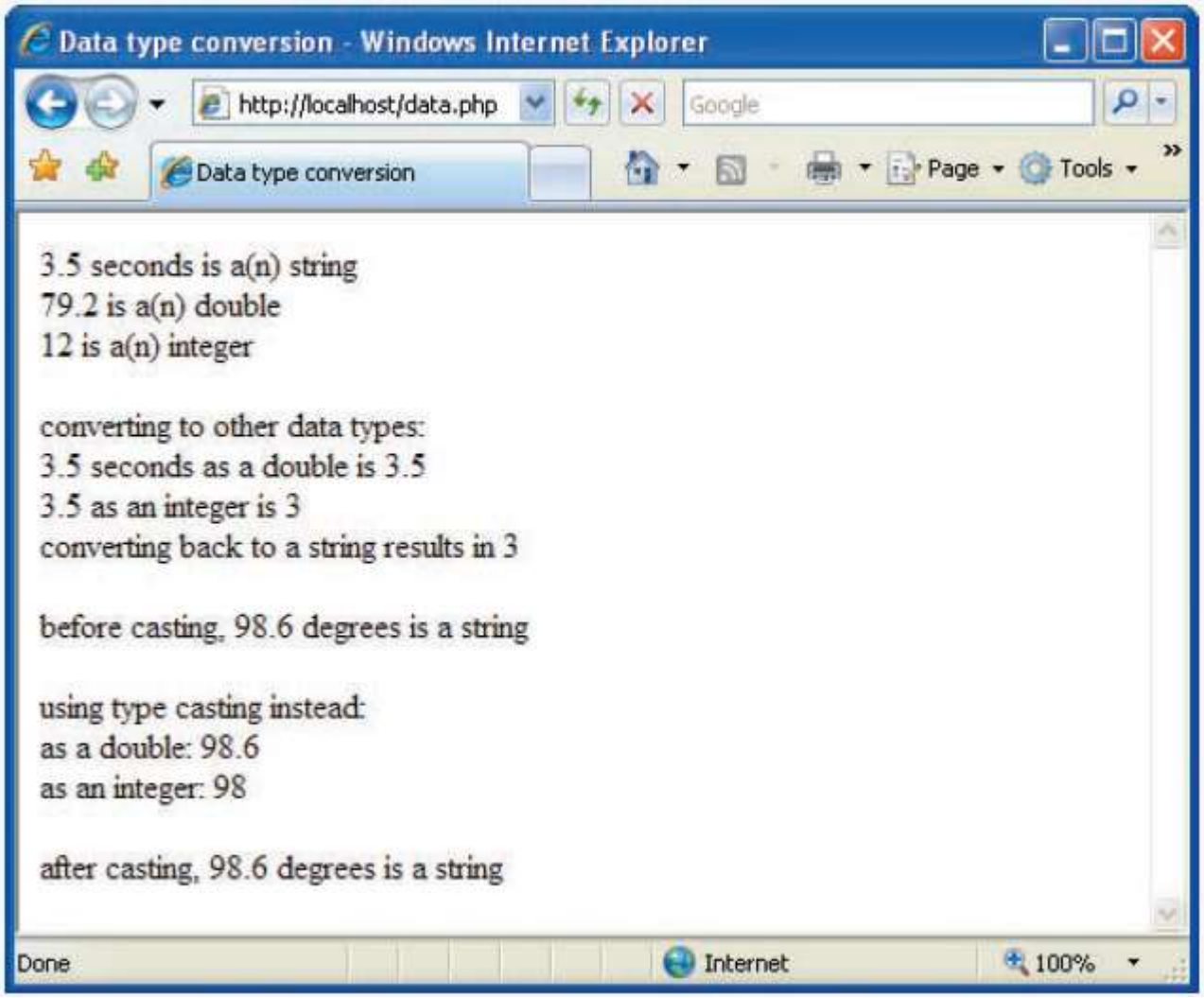
Type	Description
`int, integer`	Whole numbers (i.e., numbers without a decimal point).
`float, double, real`	Real numbers (i.e., numbers containing a decimal point).
`string`	Text enclosed in either single (' ') or double ("") quotes. [*Note:* Using double quotes allows PHP to recognize more escape sequences.]
`bool, boolean`	True or false.
`array`	Group of elements.
`object`	Group of associated data and methods.
`resource`	An external source—usually information from a database.
`NULL`	No value.

Converting Between Data Types

```php
<?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.3: data.php -->
<!-- Data type conversion. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Data type conversion</title>
   </head>
   <body>
      <?php
         // declare a string, double and integer
         $testString = "3.5 seconds";
         $testDouble = 79.2;
         $testInteger = 12;
      ?><!-- end PHP script -->

      <!-- print each variable's value and type -->
      <?php
         print( "$testString is a(n) " . gettype( $testString )
            . "<br />" );
```

```
   print( "$testDouble is a(n) " . gettype( $testDouble )
      . "<br />" );
   print( "$testInteger is a(n) " . gettype( $testInteger)
      . "<br />" );
?><!-- end PHP script -->
<br />
converting to other data types:<br />
<?php
   // call function settype to convert variable
   // testString to different data types
   print( "$testString" );
   settype( $testString, "double" );
   print( " as a double is $testString <br />" );
   print( "$testString" );
   settype( $testString, "integer" );
   print( " as an integer is $testString <br />" );
   settype( $testString, "string" );
   print( "converting back to a string results in
      $testString <br /><br />" );
```

```
        // use type casting to cast variables to a different type
        $data = "98.6 degrees";
        print( "before casting, $data is a " .
           gettype( $data ) . "<br /><br />" );
        print( "using type casting instead: <br />
           as a double: " . (double) $data .
           "<br />as an integer: " . (integer) $data );
        print( "<br /><br />after casting, $data is a " .
          gettype( $data ) );
     ?><!-- end PHP script -->
   </body>
</html>
```


Data type conversion - Windows Internet Explorer
http://localhost/data.php
Google
Data type conversion
Page
Tools
3.5 seconds is a(n) string
79.2 is a(n) double
12 is a(n) integer
converting to other data types:
3.5 seconds as a double is 3.5
3.5 as an integer is 3
converting back to a string results in 3
before casting, 98.6 degrees is a string
using type casting instead:
as a double: 98.6
as an integer: 98
after casting, 98.6 degrees is a string
Done
Internet
100%

Converting Between Data Types

- **gettype,** which returns the current type of its argument.
- **settype** to modify the type of each variable
  - settype function takes two arguments
    - the variable whose type is to be changed and the variable's new type

Converting Between Data Types

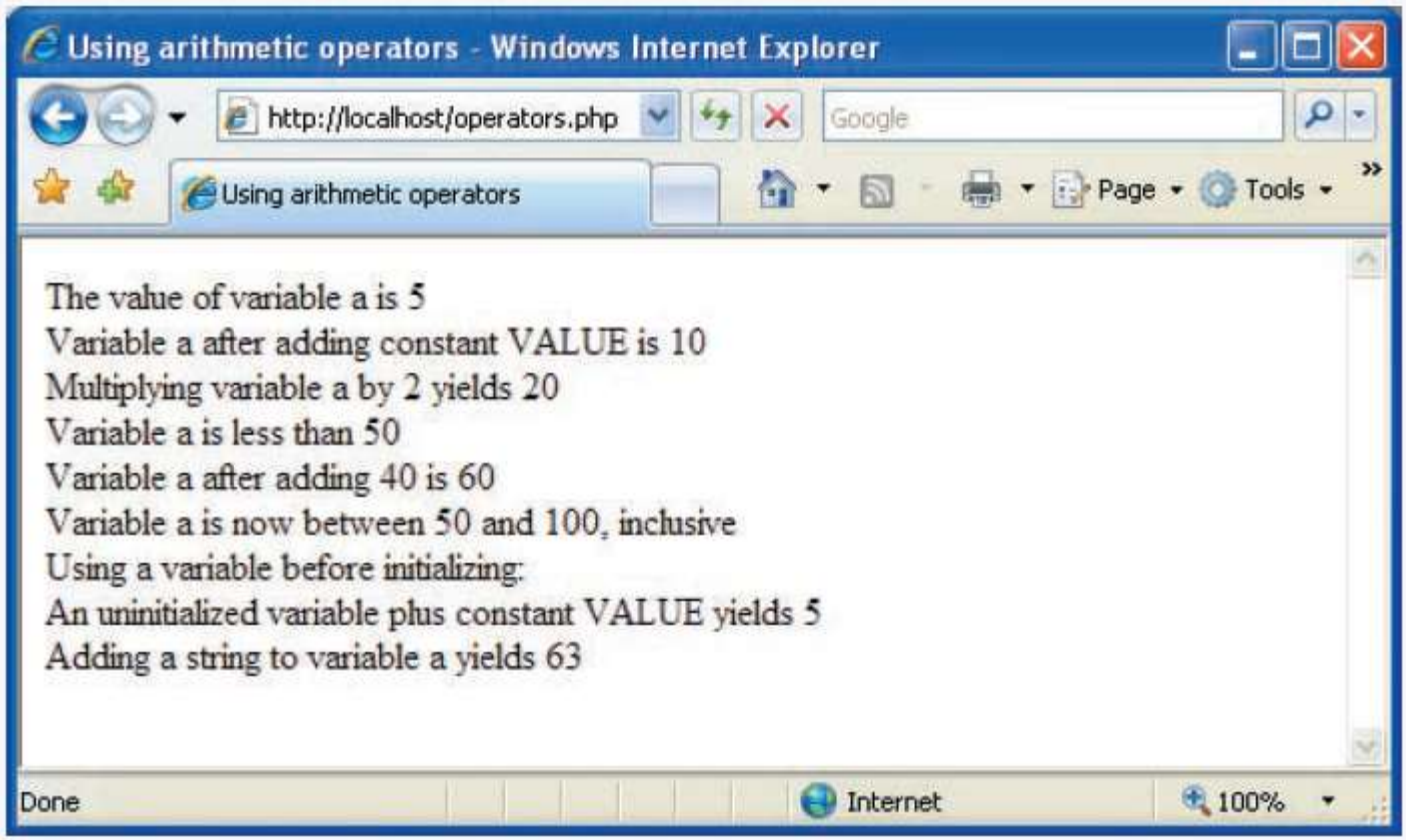
- Another option for conversion between types is **casting (or type casting).**
- Unlike settype, casting does not change a variable's content—it creates a temporary copy of a variable's value in memory
- The **concatenation operator** (.) combines multiple strings in the same print statement

Arithmetic Operators

```php
<?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.4: operators.php -->
<!-- Using arithmetic operators. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Using arithmetic operators</title>
   </head>
   <body>
      <?php
         $a = 5;
         print( "The value of variable a is $a <br />" );

         // define constant VALUE
         define( "VALUE", 5 );

         // add constant VALUE to variable $a
         $a = $a + VALUE;
         print( "Variable a after adding constant VALUE
            is $a <br />" );
```

```
// multiply variable $a by 2
$a *= 2;
print( "Multiplying variable a by 2 yields $a <br />" );

// test if variable $a is less than 50
if ( $a < 50 )
   print( "Variable a is less than 50 <br />" );

// add 40 to variable $a
$a += 40;
print( "Variable a after adding 40 is $a <br />" );

// test if variable $a is 50 or less
if ( $a < 51 )
   print( "Variable a is still 50 or less<br />" );

// test if variable $a is between 50 and 100, inclusive
elseif ( $a < 101 )
   print( "Variable a is now between 50 and 100,
      inclusive<br />" );
else
   print( "Variable a is now greater than 100 <br />" );
```

```
      // print an uninitialized variable
      print( "Using a variable before initializing:
         $nothing <br />" ); // nothing evaluates to ""

      // add constant VALUE to an uninitialized variable
      $test = $num + VALUE; // num evaluates to 0
      print( "An uninitialized variable plus constant
         VALUE yields $test <br />" );

      // add a string to an integer
      $str = "3 dollars";
      $a += $str;
      print( "Adding a string to variable a yields $a <br />" );
   ?><!-- end PHP script -->
</body>
</html>
```


Using arithmetic operators - Windows Internet Explorer
http://localhost/operators.php
Google
Using arithmetic operators
Page
Tools
The value of variable a is 5
Variable a after adding constant VALUE is 10
Multiplying variable a by 2 yields 20
Variable a is less than 50
Variable a after adding 40 is 60
Variable a is now between 50 and 100, inclusive
Using a variable before initializing:
An uninitialized variable plus constant VALUE yields 5
Adding a string to variable a yields 63
Done
Internet
100%

PHP keywords

abstract	die	exit	interface	require
and	do	extends	isset	require_once
array	echo	__FILE__	__LINE__	return
as	else	file	line	static
break	elseif	final	list	switch
case	empty	for	__METHOD__	throw
catch	enddeclare	foreach	method	try
__CLASS__	endfor	__FUNCTION__	new	unset
class	endforeach	function	or	use
clone	endif	global	php_user_filter	var
const	endswitch	if	print	while
continue	endwhile	implements	private	xor
declare	eval	include	protected	
default	exception	include_once	public	

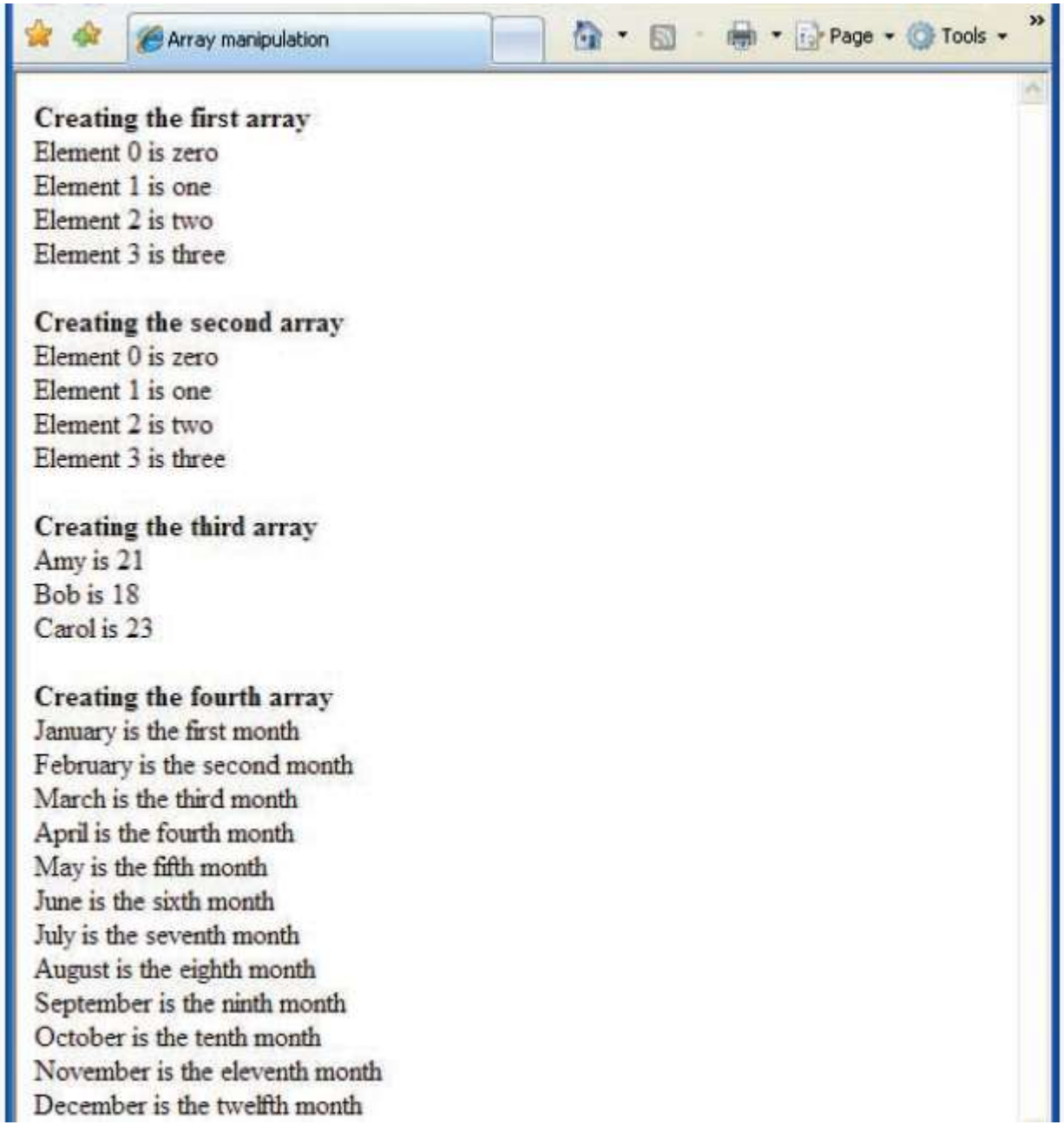
Array Manipulation

```
<?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.6: arrays.php -->
<!-- Array manipulation. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Array manipulation</title>
   </head>
   <body>
      <?php
```

```php
// create array first
print( "<strong>Creating the first array</strong><br />" );
$first[ 0 ] = "zero";
$first[ 1 ] = "one";
$first[ 2 ] = "two";
$first[] = "three";

// print each element's index and value
for ( $i = 0; $i < count( $first ); $i++ )
   print( "Element $i is $first[$i] <br />" );

print( "<br /><strong>Creating the second array
   </strong><br />" );

// call function array to create array second
$second = array( "zero", "one", "two", "three" );

for ( $i = 0; $i < count( $second ); $i++ )
   print( "Element $i is $second[$i] <br />" );

print( "<br /><strong>Creating the third array
   </strong><br />" );
```

```php
// assign values to entries using nonnumeric indices
$third[ "Amy" ] = 21;
$third[ "Bob" ] = 18;
$third[ "Carol" ] = 23;

// iterate through the array elements and print each
// element's name and value
for ( reset( $third ); $element = key( $third ); next( $third ) )
   print( "$element is $third[$element] <br />" );

print( "<br /><strong>Creating the fourth array
   </strong><br />" );

// call function array to create array fourth using
// string indices
$fourth = array(
   "January"   => "first",   "February" => "second",
   "March"     => "third",   "April"    => "fourth",
   "May"       => "fifth",   "June"     => "sixth",
   "July"      => "seventh", "August"   => "eighth",
   "September" => "ninth",   "October"  => "tenth",
   "November"  => "eleventh","December" => "twelfth"
   );
```

```
         // print each element's name and value
         foreach ( $fourth as $element => $value )
            print( "$element is the $value month <br />" );
      ?><!-- end PHP script -->
   </body>
</html>
```


Array manipulation
Page
Tools
Creating the first array
Element 0 is zero
Element 1 is one
Element 2 is two
Element 3 is three
Creating the second array
Element 0 is zero
Element 1 is one
Element 2 is two
Element 3 is three
Creating the third array
Amy is 21
Bob is 18
Carol is 23
Creating the fourth array
January is the first month
February is the second month
March is the third month
April is the fourth month
May is the fifth month
June is the sixth month
July is the seventh month
August is the eighth month
September is the ninth month
October is the tenth month
November is the eleventh month
December is the twelfth month

Array Manipulation

- Arrays can have float or nonnumeric indices
- An array with noninteger indices is called an **associative array**
- PHP provides functions for iterating through the elements of an array
- Each array has a built-in **internal pointer,** which points to the array element currently being referenced.

Array Manipulation

- Function **reset** sets the internal pointer to the first array element.
- Function **key** returns the index of the element currently referenced by the internal pointer,
- Function **next** moves the internal pointer to the next element and returns the element.

Array Manipulation

- To override the automatic numeric indexing performed by function array, you can use operator =>
- The value to the left of the operator is the array index and the value to the right is the element’s value

Array Manipulation

- The **foreach** control statement is specifically designed for iterating through arrays, especially associative arrays
- The foreach statement starts with the array to iterate through, followed by the keyword **as**, followed by two variables—the first is assigned the index of the element, and the second is assigned the value of that index

String Processing and Regular Expressions

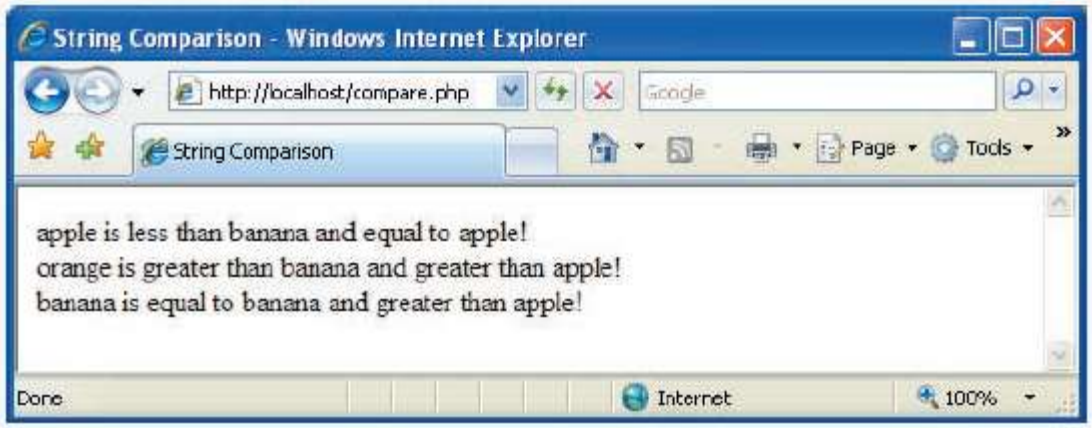
- Text manipulation is usually done with **regular expressions**
  - a series of characters that serve as pattern-matching templates in strings, text files and databases

Comparing Strings

- Many string-processing tasks can be accomplished by using the **equality** and **comparison** operators

```php
<?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.7: compare.php -->
<!-- Using the string-comparison operators. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>String Comparison</title>
   </head>
   <body>
      <?php
         // create array fruits
         $fruits = array( "apple", "orange", "banana" );

         // iterate through each array element
         for ( $i = 0; $i < count( $fruits ); $i++ )
         {
            // call function strcmp to compare the array element
            // to string "banana"
            if ( strcmp( $fruits[ $i ], "banana" ) < 0 )
               print( $fruits[ $i ] . " is less than banana " );
            elseif ( strcmp( $fruits[ $i ], "banana" ) > 0 )
               print( $fruits[ $i ] . " is greater than banana " );
            else
               print( $fruits[ $i ] . " is equal to banana " );

            // use relational operators to compare each element
            // to string "apple"
```

```
            if ( $fruits[ $i ] < "apple" )
               print( "and less than apple! <br />" );
            elseif ( $fruits[ $i ] > "apple" )
               print( "and greater than apple! <br />" );
            elseif ( $fruits[ $i ] == "apple" )
               print( "and equal to apple! <br />" );
         } // end for
      ?><!-- end PHP script -->
   </body>
</html>
```



Comparing Strings

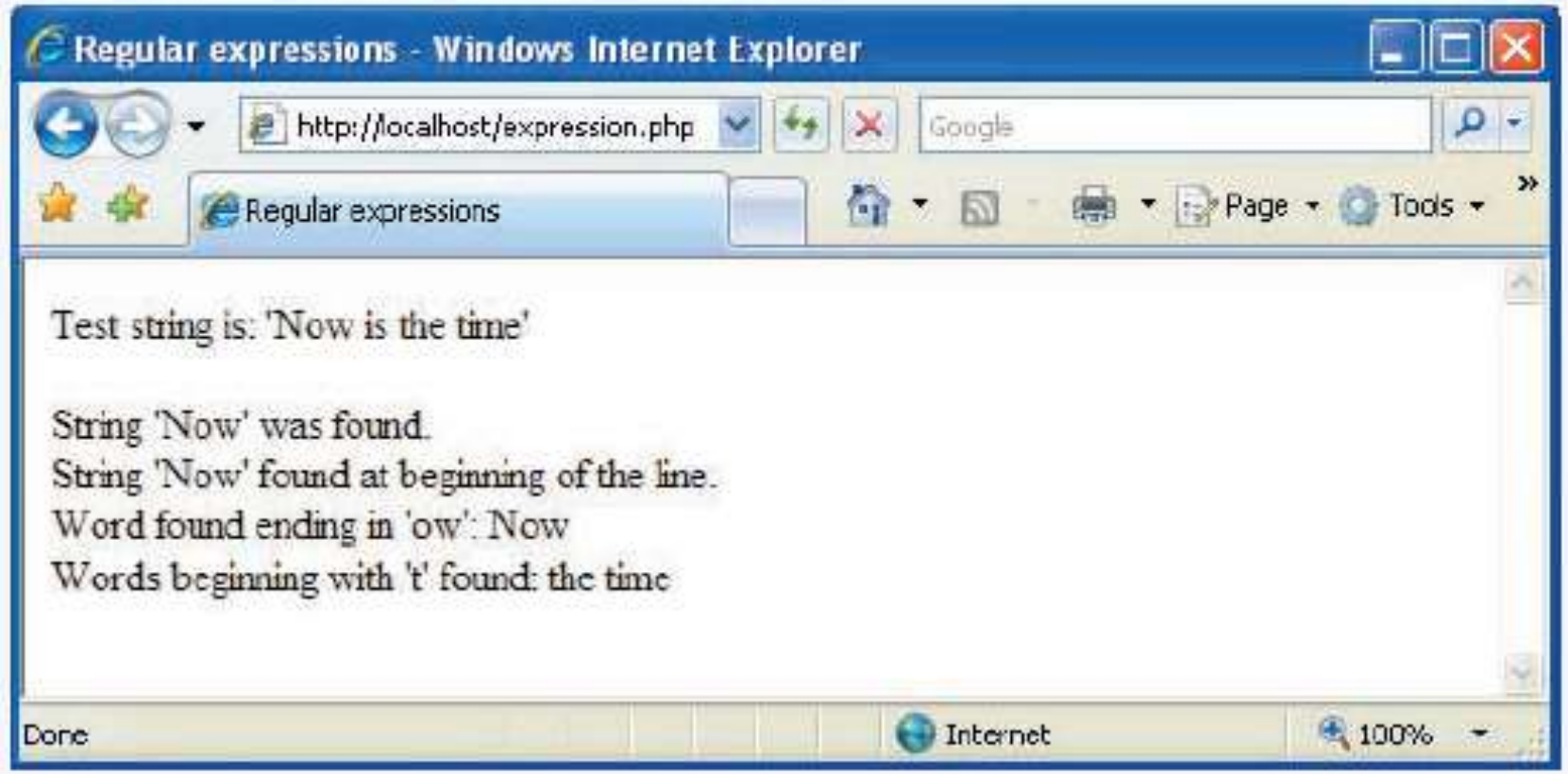
- Function **strcmp** compares two strings and returns -1 if the first string alphabetically precedes the second string, 0 if the strings are equal, and 1 if the first string alphabetically follows the second
- Relational operators (==, !=, <, <=, > and >=) can also be used to compare strings.

Regular Expressions

- Functions **ereg** and **preg_match** use regular expressions to search a string for a specified pattern.
- Function **ereg** recognizes **Portable Operating System Interface (POSIX)** extended regular expressions
- Function **preg_match provides Perl-compatible regular expressions (PCRE)**
- To use preg_match, you must install the PCRE library on your web server and add support for the library to PHP

```php
<?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.8: expression.php -->
<!-- Regular expressions. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
   <title>Regular expressions</title>
</head>
<body>
   <?php
      $search = "Now is the time";
      print( "Test string is: '$search'<br /><br />" );

      // call ereg to search for pattern 'Now' in variable search
      if ( ereg( "Now", $search ) )
         print( "String 'Now' was found.<br />" );

      // search for pattern 'Now' in the beginning of the string
      if ( ereg( "^Now", $search ) )
         print( "String 'Now' found at beginning
            of the line.<br />" );
```

```
         // search for pattern 'Now' at the end of the string
         if ( ereg( "Now$", $search ) )
            print( "String 'Now' was found at the end
               of the line.<br />" );

         // search for any word ending in 'ow'
         if ( ereg( "[[:<:]]([a-zA-Z]*ow)[[:>:]]", $search, $match ) )
            print( "Word found ending in 'ow': " .
               $match[ 1 ] . "<br />" );

         // search for any words beginning with 't'
         print( "Words beginning with 't' found: ");

         while ( eregi( "[[:<:]](t[[:alpha:]]+)[[:>:]]",
            $search, $match ) )
         {
            print( $match[ 1 ] . " " );

            // remove the first occurrence of a word beginning
            // with 't' to find other instances in the string
            $search = ereg_replace( $match[ 1 ], "", $search );
         } // end while
      ?><!-- end PHP script -->
   </body>
</html>
```


Regular expressions - Windows Internet Explorer
http://localhost/expression.php
Google
Regular expressions
Page
Tools
Test string is: 'Now is the time'
String 'Now' was found.
String 'Now' found at beginning of the line.
Word found ending in 'ow': Now
Words beginning with 't' found: the time
Done
Internet
100%

Regular Expressions

- Anything enclosed in single quotes is not interpolated (unless the single quotes are nested in a double-quoted string literal)
- Function ereg takes two arguments
  - a regular expression pattern to search for and the string to search
- Case mixture is significant in patterns
- PHP provides function **eregi** for specifying case-insensitive pattern matches

Regular Expressions

- regular expressions can include **metacharacters** that specify patterns
- metacharacters include the ^, $ and . characters
- The **caret (^)** metacharacter matches the beginning of a string
- The **dollar sign ($)** matches the end of a string
- The **period (.)** metacharacter matches any single character

Regular Expressions

- **Bracket expressions** are lists of characters enclosed in square brackets ([]) that match any single character from the list
- Ranges can be specified by supplying the beginning and the end of the range separated by a **dash (-)**
- The bracket expression **[a-z]** matches any lowercase letter and **[A-Z]** matches any uppercase letter.

Regular Expressions

- The special bracket expressions **[[:<:]]** and **[[:>:]]** match the beginning and end of a word, respectively

Some PHP quantifiers

Quantifier	Matches
{*n*}	Exactly *n* times.
{*m*,*n*}	Between *m* and *n* times, inclusive.
{*n*,}	*n* or more times.
+	One or more times (same as {1,}).
*	Zero or more times (same as {0,}).
?	Zero or one time (same as {0,1}).

Some PHP quantifiers

Finding Matches

- The optional third argument to function **ereg** is an array that stores matches to the regular expression

Character Classes

- Character classes are enclosed by the delimiters [: and :].

Some PHP character classes

Character class	Description
`alnum`	Alphanumeric characters (i.e., letters [a-zA-Z] or digits [0-9]).
`alpha`	Word characters (i.e., letters [a-zA-Z]).
`digit`	Digits.
`space`	White space.
`lower`	Lowercase letters.
`upper`	Uppercase letters.

Some PHP character classes

- When the expression is placed in another set of brackets, such as [[:alpha:]], it is a regular expression matching a single character that is a member of the class
- A bracketed expression containing two or more adjacent character classes in the class delimiters represents those character sets combined.
- Eg.: the expression [[:upper:][:lower:]]* represents all strings of uppercase and lowercase letters in any order

Some PHP character classes

- The expression [[:upper:]][[:lower:]]* matches strings with a single uppercase letter followed by any number of lowercase characters.
- The expression ([[:upper:]][[:lower:]])* is an expression for all strings that alternate between uppercase and lowercase characters (starting with uppercase and ending with lowercase).

Form Processing and Business Logic

Superglobal Arrays

- Superglobal arrays are associative arrays predefined by PHP that hold variables acquired from user input, the environment or the web server, and are accessible in any variable scope

Some superglobal arrays

Variable name	Description
$_SERVER	Data about the currently running server.
$_ENV	Data about the client's environment.
$_GET	Data sent to the server by a get request.
$_POST	Data sent to the server by a post request.
$_COOKIE	Data contained in cookies on the client's computer.
$GLOBALS	Array containing all global variables.

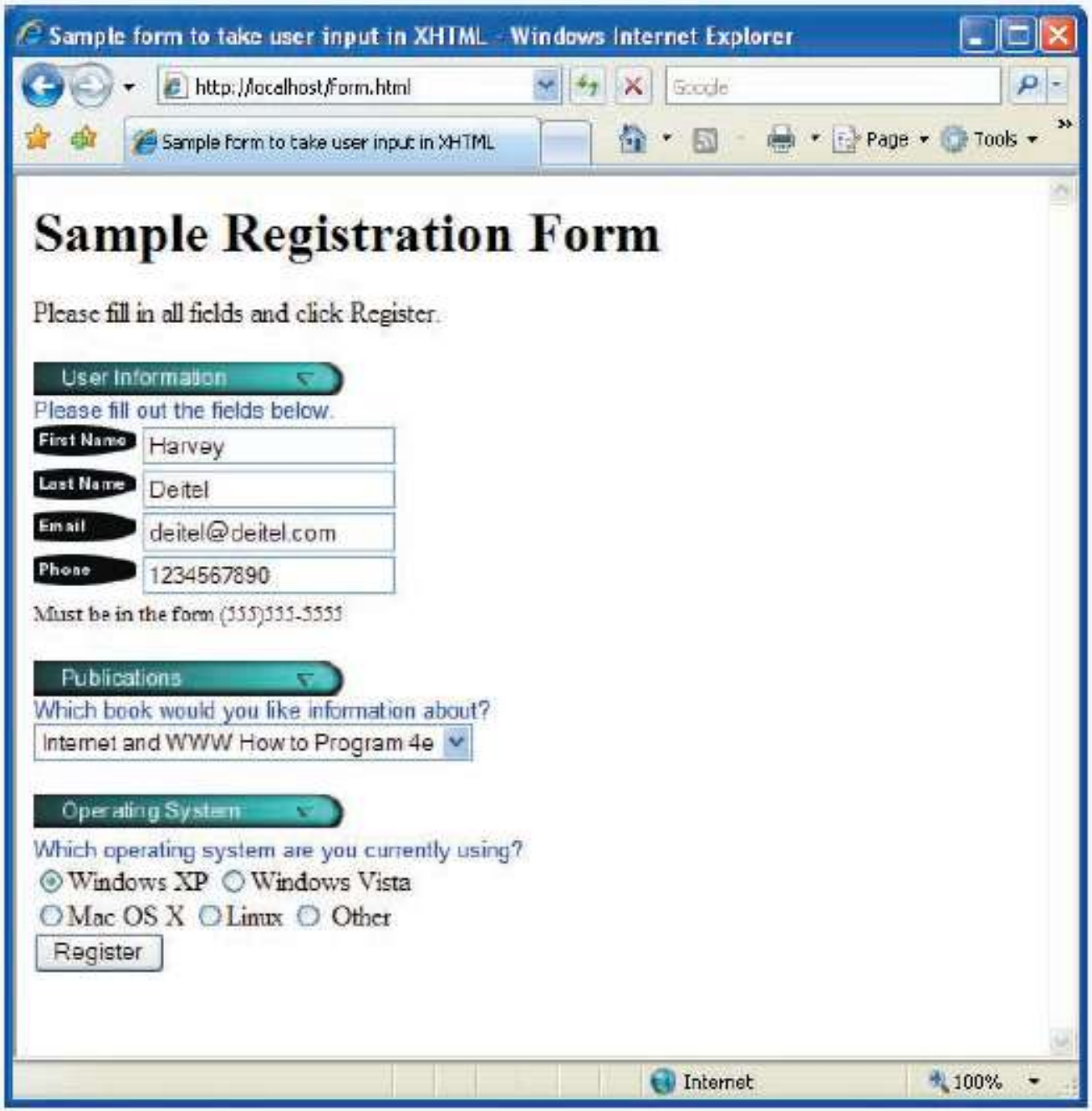
Using PHP to Process XHTML Forms

- XHTML forms enable web pages to collect data from users and send it to a web server for processing

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.12: form.html -->
<!-- XHTML form for gathering user input. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Sample form to take user input in XHTML</title>
      <style type = "text/css">
         .prompt { color: blue;
                   font-family: sans-serif;
                   font-size: smaller }
      </style>
   </head>
   <body>
      <h1>Sample Registration Form</h1>
      <p>Please fill in all fields and click Register.</p>

      <!-- post form data to form.php -->
      <form method = "post" action = "form.php">
         <div>
            <img src = "images/user.gif" alt = "User" /><br />
            <span class = "prompt">
               Please fill out the fields below.<br />
            </span>
```

```
<!-- create four text boxes for user input -->
<img src = "images/fname.gif" alt = "First Name" />
<input type = "text" name = "fname" /><br />

<img src = "images/lname.gif" alt = "Last Name" />
<input type = "text" name = "lname" /><br />

<img src = "images/email.gif" alt = "Email" />
<input type = "text" name = "email" /><br />

<img src = "images/phone.gif" alt = "Phone" />
<input type = "text" name = "phone" /><br />

<span style = "font-size: 10pt">
   Must be in the form (555)555-5555</span>
<br /><br />

<img src = "images/downloads.gif"
alt = "Publications" /><br />

<span class = "prompt">
   Which book would you like information about?
</span><br />

<!-- create drop-down list containing book names -->
<select name = "book">
```

```
            <option>Internet and WWW How to Program 4e</option>
            <option>C++ How to Program 6e</option>
            <option>Java How to Program 7e</option>
            <option>Visual Basic 2005 How to Program 3e</option>
         </select>
         <br /><br />

         <img src = "images/os.gif" alt = "Operating System" />
         <br /><span class = "prompt">
            Which operating system are you currently using?
         <br /></span>

         <!-- create five radio buttons -->
         <input type = "radio" name = "os" value = "Windows XP"
            checked = "checked" /> Windows XP
         <input type = "radio" name = "os" value =
            "Windows Vista" /> Windows Vista<br />
         <input type = "radio" name = "os" value =
            "Mac OS X" /> Mac OS X
         <input type = "radio" name = "os" value = "Linux" /> Linux
         <input type = "radio" name = "os" value = "Other" />
            Other<br />

         <!-- create a submit button -->
         <input type = "submit" value = "Register" />
      </div>
   </form>
</body>
</html>
```

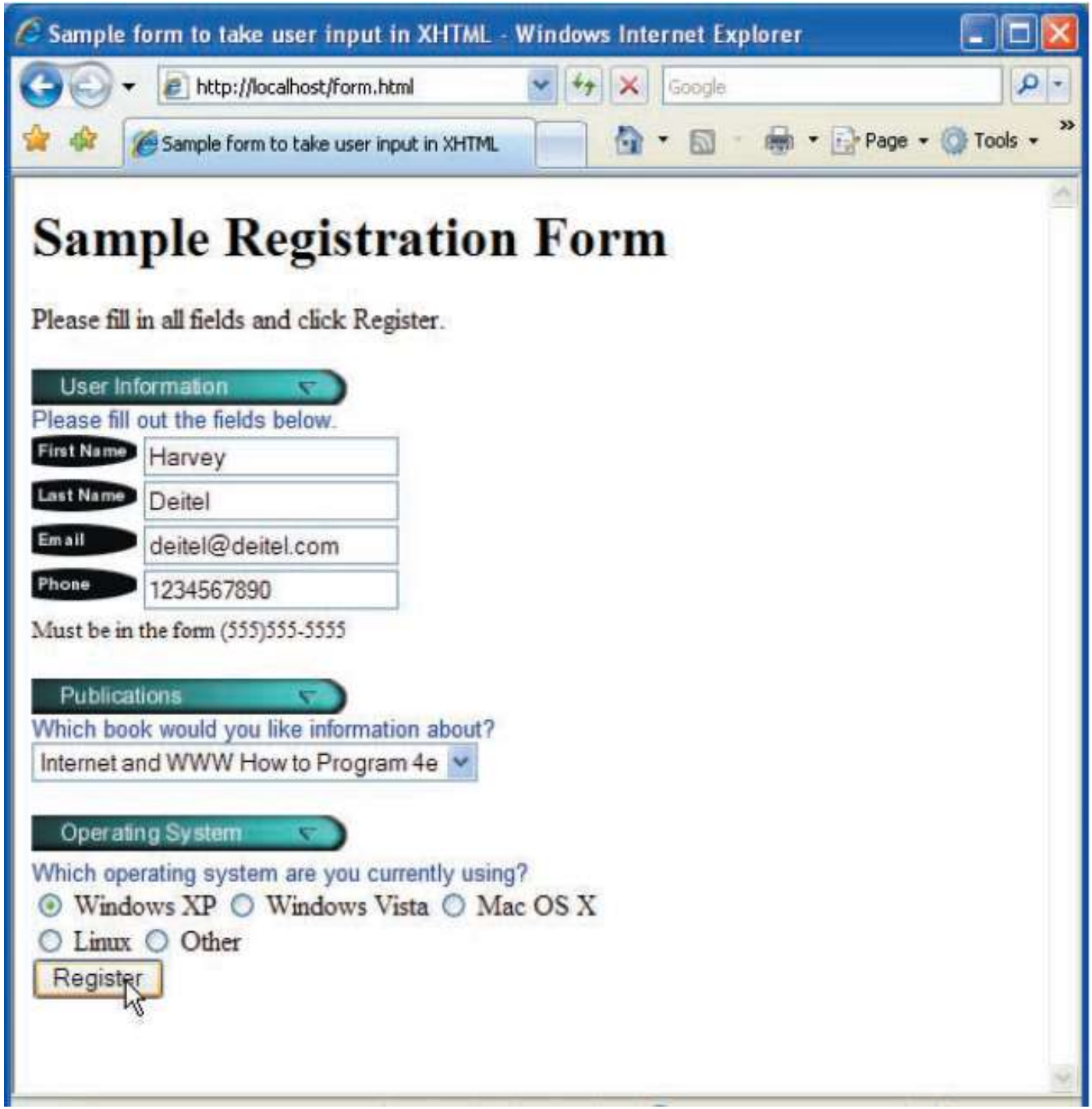

Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost/Form.html
Sample form to take user input in XHTML
Page
Tools
Sample Registration Form
Please fill in all fields and click Register.
User Information
Please fill out the fields below.
First Name
Harvey
Last Name
Deitel
Email
deitel@deitel.com
Phone
1234567890
Must be in the form (555)555-5555
Publications
Which book would you like information about?
Internet and WWW How to Program 4e
Operating System
Which operating system are you currently using?
Windows XP
Windows Vista
Mac OS X
Linux
Other
Register
Internet
100%

```
<?php print( '<?xml version = "1.0" encoding = "utf-8"?>' ) ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- Fig. 23.13: form.php -->
<!-- Process information sent from form.html. -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Form Validation</title>
      <style type = "text/css">
         body      { font-family: arial, sans-serif }
         div       { font-size: 10pt;
                     text-align: center }
         table     { border: 0 }
         td        { padding-top: 2px;
                     padding-bottom: 2px;
                     padding-left: 10px;
                     padding-right: 10px }
         .error    { color: red }
         .distinct { color: blue }
         .name     { background-color: #ffffaa }
```

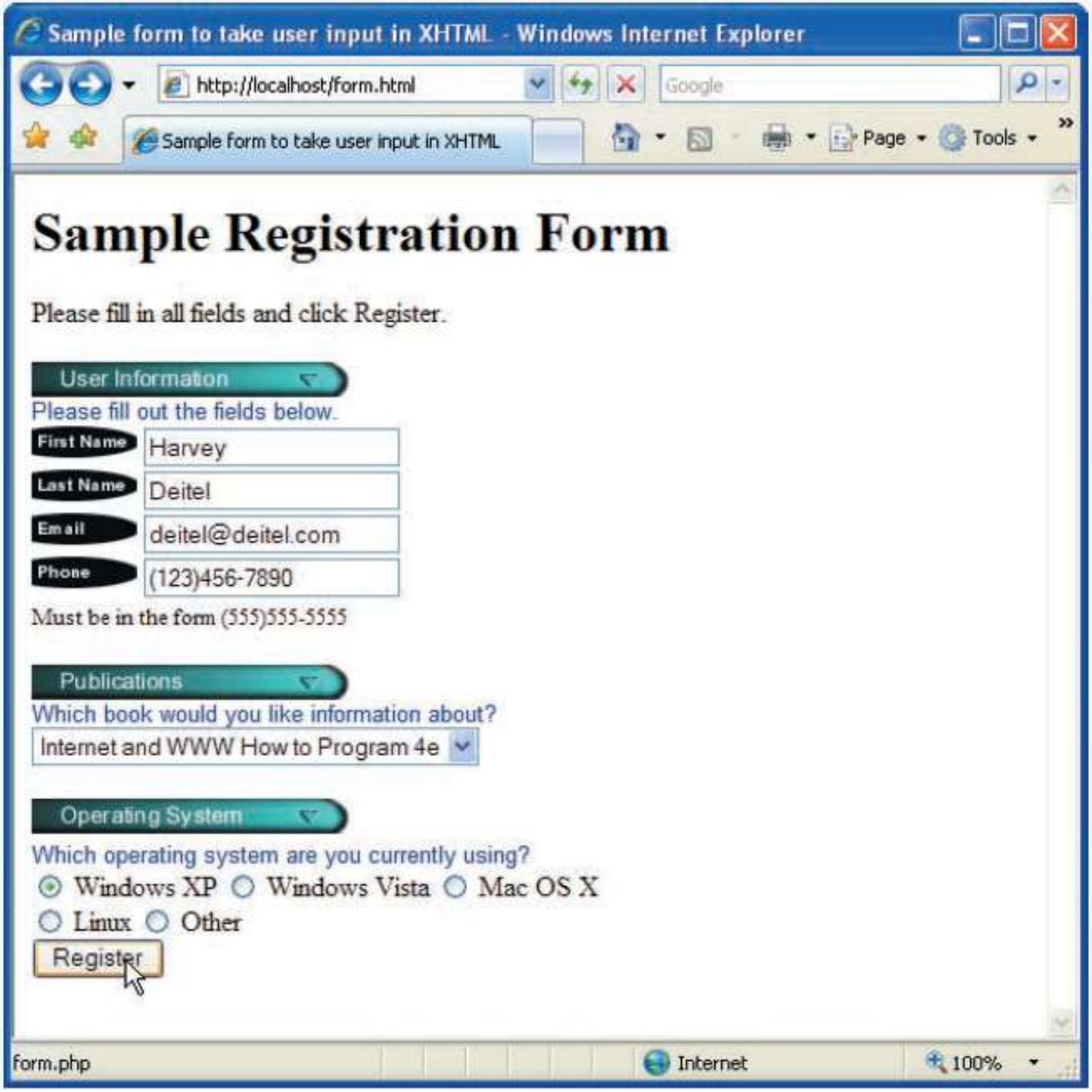
```
      .email    { background-color: #ffffbb }
      .phone    { background-color: #ffffcc }
      .os       { background-color: #ffffdd }
   </style>
</head>
<body>
   <?php
      extract( $_POST );

      // determine whether phone number is valid and print
      // an error message if not
      if ( !ereg( "^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$", $phone ) )
      {
         print( "<p><span class = 'error'>
            Invalid phone number</span><br />
            A valid phone number must be in the form
            <strong>(555)555-5555</strong><br />
            <span class = 'distinct'>
            Click the Back button, enter a valid phone
            number and resubmit.<br /><br />
            Thank You.</span></p>" );
         die( "</body></html>" ); // terminate script execution
      }
   ?><!-- end PHP script -->
```

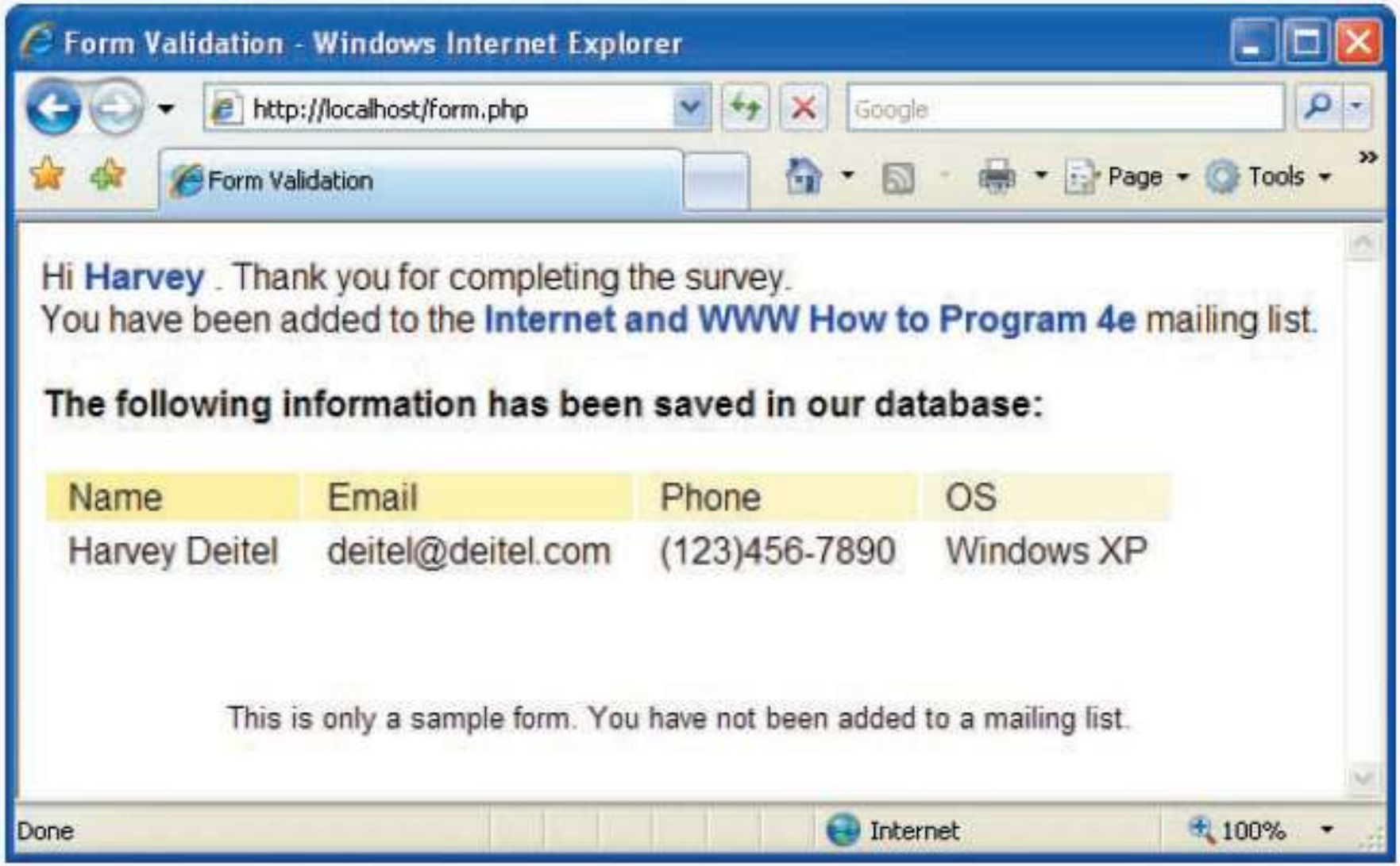
```
<p>Hi
   <span class = "distinct">
      <strong><?php print( "$fname" ); ?></strong>
   </span>.
   Thank you for completing the survey.<br />
   You have been added to the
   <span class = "distinct">
      <strong><?php print( "$book " ); ?></strong>
   </span>
   mailing list.
</p>
<p><strong>The following information has been saved
    in our database:</strong></p>
<table>
   <tr>
      <td class = "name">Name </td>
      <td class = "email">Email</td>
      <td class = "phone">Phone</td>
      <td class = "os">OS</td>
   </tr>
   <tr>
      <?php
         // print each form field's value
         print( "<td>$fname $lname</td>
            <td>$email</td>
            <td>$phone</td>
            <td>$os</td>" );
      ?><!-- end PHP script -->
   </tr>
```

```
        </table>
        <br /><br /><br />
        <div>This is only a sample form.
           You have not been added to a mailing list.</div>
    </body>
</html>
```


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost/form.html
Google
Sample form to take user input in XHTML
Page
Tools
Sample Registration Form
Please fill in all fields and click Register.
User Information
Please fill out the fields below.
First Name
Harvey
Last Name
Deitel
Email
deitel@deitel.com
Phone
1234567890
Must be in the form (555)555-5555
Publications
Which book would you like information about?
Internet and WWW How to Program 4e
Operating System
Which operating system are you currently using?
Windows XP
Windows Vista
Mac OS X
Linux
Other
Register

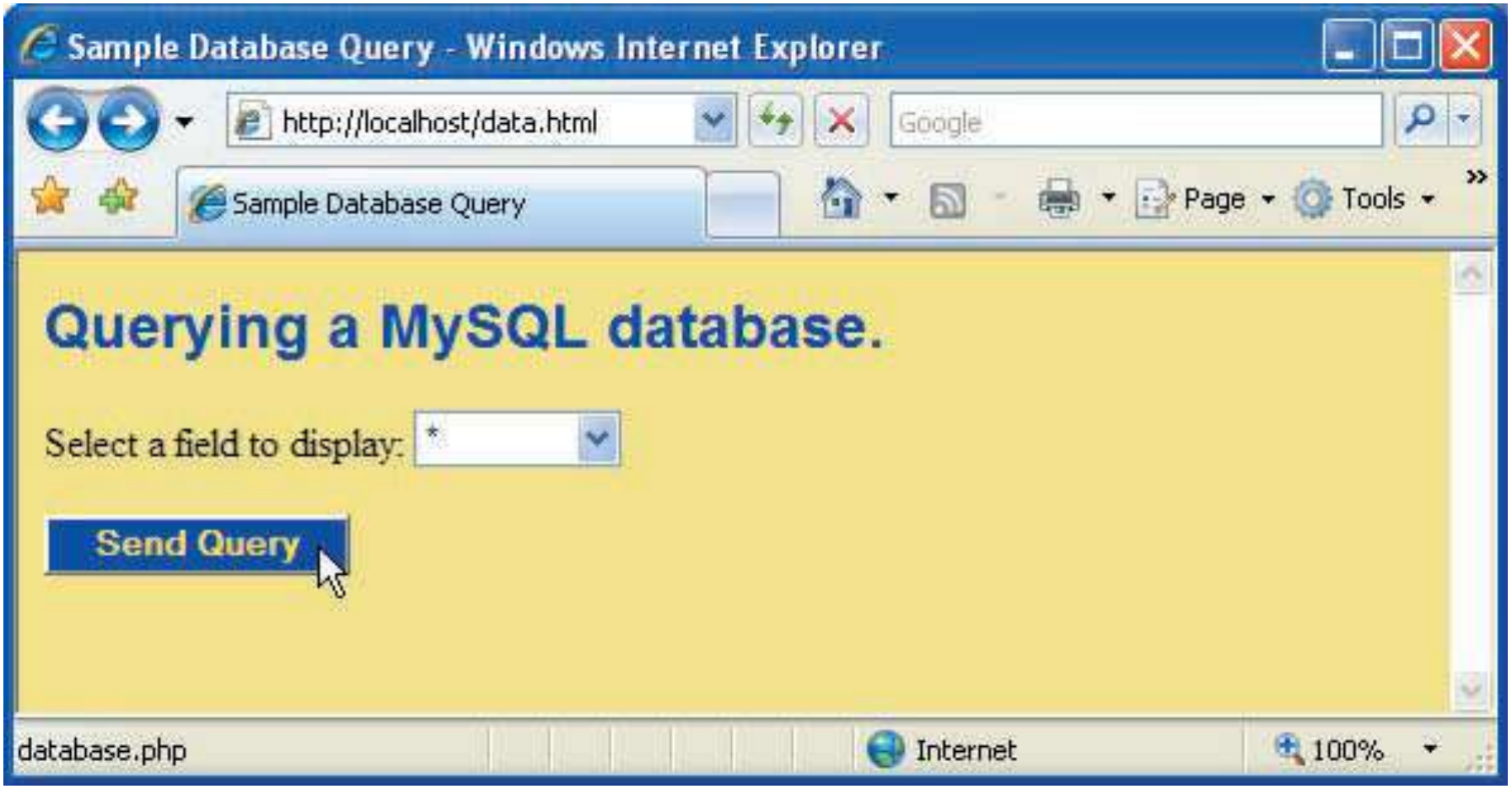

Form Validation - Windows Internet Explorer
http://localhost/form.php
Google
Form Validation
Page
Tools
Invalid phone number
A valid phone number must be in the form (555)555-5555
Click the Back button, enter a valid phone number and resubmit.
Thank You.
Done
Internet
100%


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost/form.html
Google
Sample form to take user input in XHTML
Page
Tools
Sample Registration Form
Please fill in all fields and click Register.
User Information
Please fill out the fields below.
First Name
Harvey
Last Name
Deitel
Email
deitel@deitel.com
Phone
(123)456-7890
Must be in the form (555)555-5555
Publications
Which book would you like information about?
Internet and WWW How to Program 4e
Operating System
Which operating system are you currently using?
Windows XP
Windows Vista
Mac OS X
Linux
Other
Register
form.php
Internet
100%


Form Validation - Windows Internet Explorer
http://localhost/form.php
Google
Form Validation
Page
Tools
Hi Harvey . Thank you for completing the survey.
You have been added to the Internet and WWW How to Program 4e mailing list.
The following information has been saved in our database:
Name
Email
Phone
OS
Harvey Deitel
deitel@deitel.com
(123)456-7890
Windows XP
This is only a sample form. You have not been added to a mailing list.
Done
Internet
100%

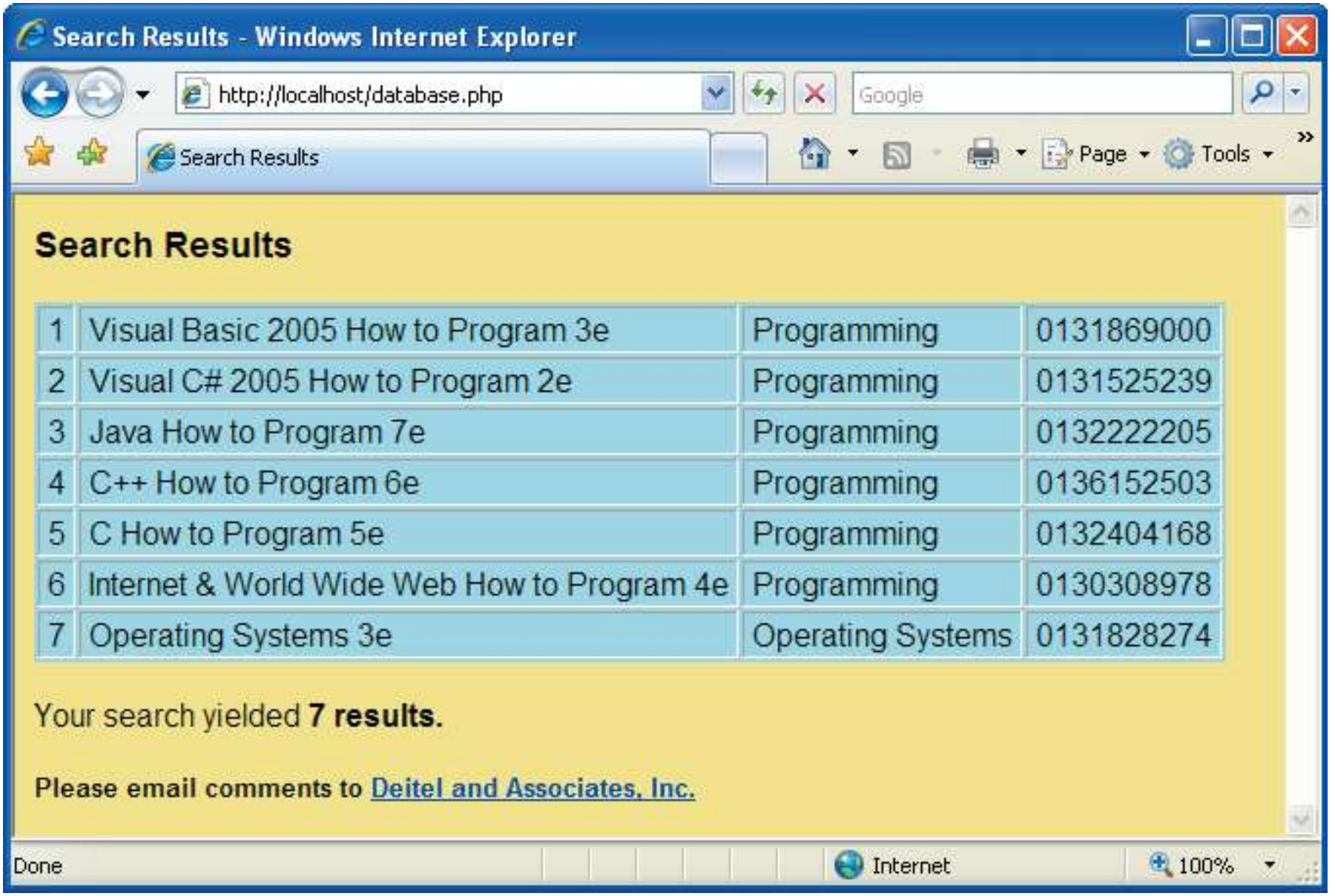
Connecting to a Database

```
<body>
   <h2> Querying a MySQL database.</h2>
   <form method = "post" action = "database.php">
      <div>
         <p>Select a field to display:
            <!-- add a select box containing options -->
            <!-- for SELECT query -->
            <select name = "select">
               <option selected = "selected">*</option>
               <option>ID</option>
               <option>Title</option>
               <option>Category</option>
               <option>ISBN</option>
            </select></p>
         <input type = "submit" value = "Send Query" />
      </div>
   </form>
</body>
```


Sample Database Query - Windows Internet Explorer
http://localhost/data.html
Google
Sample Database Query
Page
Tools
Querying a MySQL database.
Select a field to display:
*
Send Query
database.php
Internet
100%

```
<body>
   <?php
      extract( $_POST );

      // build SELECT query
      $query = "SELECT " . $select . " FROM books";

      // Connect to MySQL
      if ( !( $database = mysql_connect( "localhost",
         "iw3htp4", "iw3htp4" ) ) )
         die( "Could not connect to database </body></html>" );

      // open Products database
      if ( !mysql_select_db( "products", $database ) )
         die( "Could not open products database </body></html>" );

      // query Products database
      if ( !( $result = mysql_query( $query, $database ) ) )
      {
         print( "Could not execute query! <br />" );
         die( mysql_error() . "</body></html>" );
      } // end if

      mysql_close( $database );
   ?><!-- end PHP script -->
```

```
<h3>Search Results</h3>
<table>
   <?php
      // fetch each record in result set
      for ( $counter = 0; $row = mysql_fetch_row( $result );
         $counter++ )
      {
         // build table to display results
         print( "<tr>" );

         foreach ( $row as $key => $value )
            print( "<td>$value</td>" );

         print( "</tr>" );
      } // end for
   ?><!-- end PHP script -->
</table>
<br />Your search yielded <strong>
<?php print( "$counter" ) ?> results.<br /><br /></strong>
<h5>Please email comments to
   <a href = "mailto:deitel@deitel.com">
      Deitel and Associates, Inc.</a>
</h5>
</body>
```


Search Results - Windows Internet Explorer
http://localhost/database.php
Google
Search Results
Page
Tools
Search Results
1 Visual Basic 2005 How to Program 3e Programming 0131869000
2 Visual C# 2005 How to Program 2e Programming 0131525239
3 Java How to Program 7e Programming 0132222205
4 C++ How to Program 6e Programming 0136152503
5 C How to Program 5e Programming 0132404168
6 Internet & World Wide Web How to Program 4e Programming 0130308978
7 Operating Systems 3e Operating Systems 0131828274
Your search yielded 7 results.
Please email comments to Deitel and Associates, Inc.
Done
Internet
100%

Using Cookies

- A **cookie** is a piece of information stored in a text file on a client's computer to maintain information about the client during and between browsing sessions.
- A website can store a cookie on a client's computer to record user preferences and other information that the website can retrieve during the client's subsequent visits
- A server can access only the cookies that it has placed on the client

Using Cookies

- The information stored in a cookie is sent back to the web server from which it originated whenever the user requests a web page from that particular server

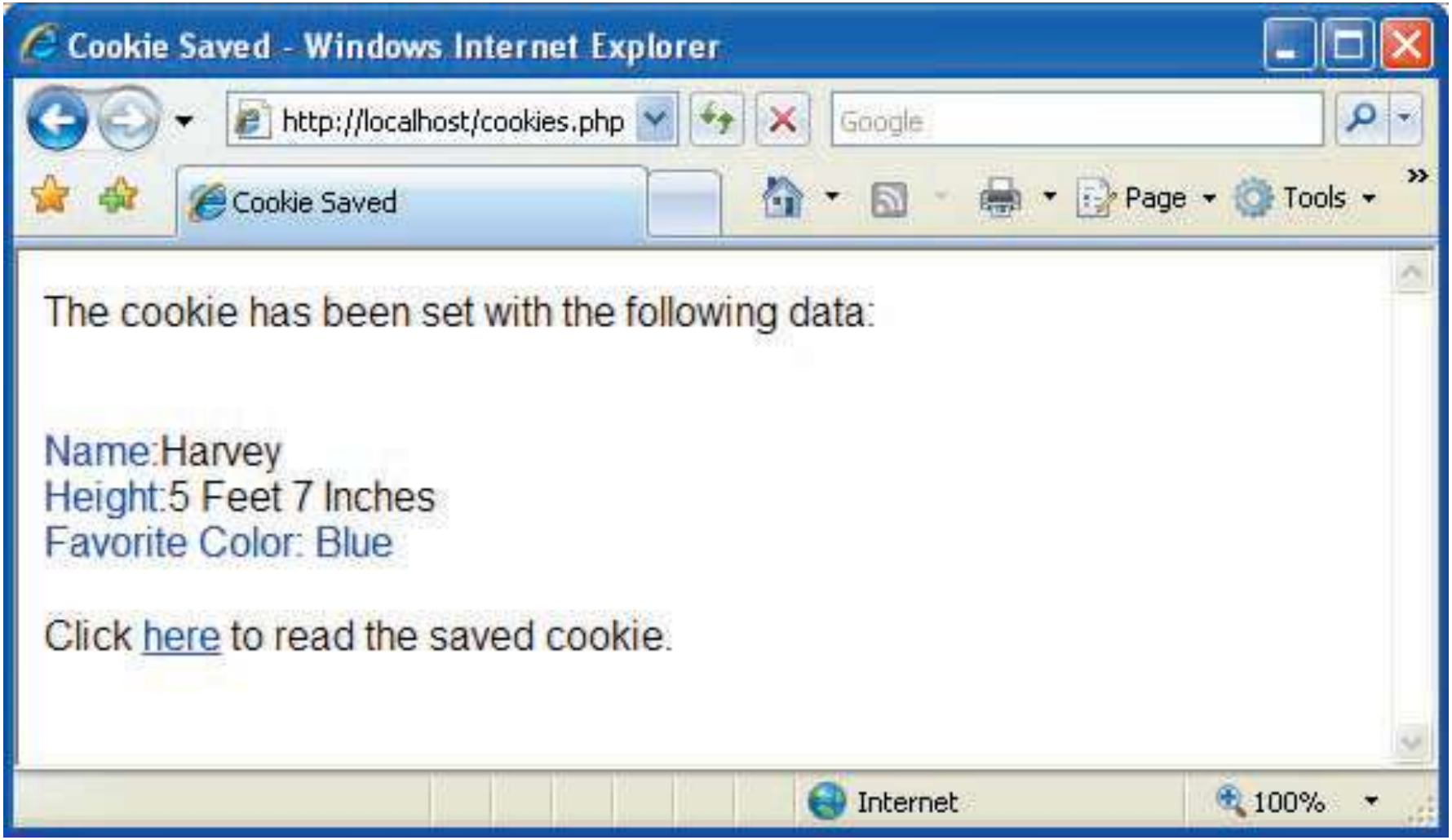
Writing Cookies

```
<body>
   <h2>Click Write Cookie to save your cookie data.</h2>
   <form method = "post" action = "cookies.php">
      <div>
         <strong>Name:</strong><br />
         <input type = "text" name = "Name" /><br />

         <strong>Height:</strong><br />
         <input type = "text" name = "Height" /><br />

         <strong>Favorite Color:</strong><br />
         <input type = "text" name = "Color" /><br />

         <input type = "submit" value = "Write Cookie"
            class = "submit" />
      </div>
   </form>
</body>
```


Writing a cookie to the client computer - Windows Internet Explorer
http://localhost/cookies.html
Google
Writing a cookie to the client computer
Page
Tools
Click Write Cookie to save your cookie data.
Name:
Harvey
Height:
5 Feet 7 Inches
Favorite Color:
Blue
Write Cookie
cookies.php
Internet
100%

```php
<?php
   // Fig. 23.17: cookies.php
   // Writing a cookie to the client.
   extract( $_POST );

   // write each form field's value to a cookie and set the
   // cookie's expiration date
   setcookie( "Name", $Name, time() + 60 * 60 * 24 * 5 );
   setcookie( "Height", $Height, time() + 60 * 60 * 24 * 5 );
   setcookie( "Color", $Color, time() + 60 * 60 * 24 * 5 );
?><!-- end PHP script -->
```

```
<body>
   <p>The cookie has been set with the following data:</p>

   <!-- print each form field's value -->
   <br /><span>Name:</span><?php print( $Name ) ?><br />
   <span>Height:</span><?php print( $Height ) ?><br />
   <span>Favorite Color:</span>
   <span style = "color: <?php print( "$Color\">$Color" ) ?>
   </span><br />
   <p>Click <a href = "readCookies.php">here</a>
      to read the saved cookie.</p>
</body>
```

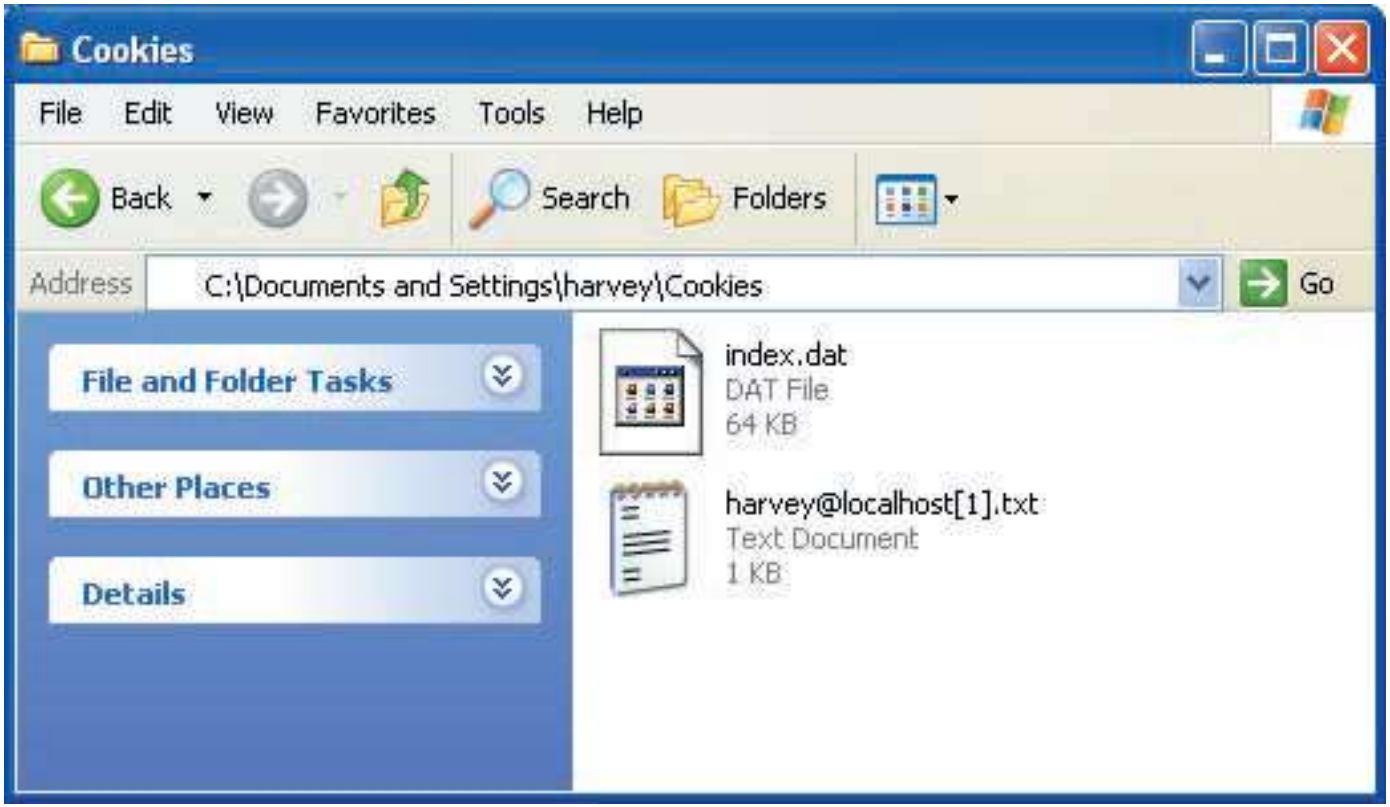

Cookie Saved - Windows Internet Explorer
http://localhost/cookies.php
Google
Cookie Saved
Page
Tools
The cookie has been set with the following data:
Name:Harvey
Height:5 Feet 7 Inches
Favorite Color: Blue
Click here to read the saved cookie.
Internet
100%

Using Cookies

- If no expiration date is specified, the cookie lasts only until the end of the current session, which is the total time until the user closes the browser. This type of cookie is known as a **session cookie,** while one with an expiration date is a **persistent cookie.**

Using Cookies

- When using Internet Explorer, cookies are stored in a **Cookies** directory on the client's machine



Reading an Existing Cookie

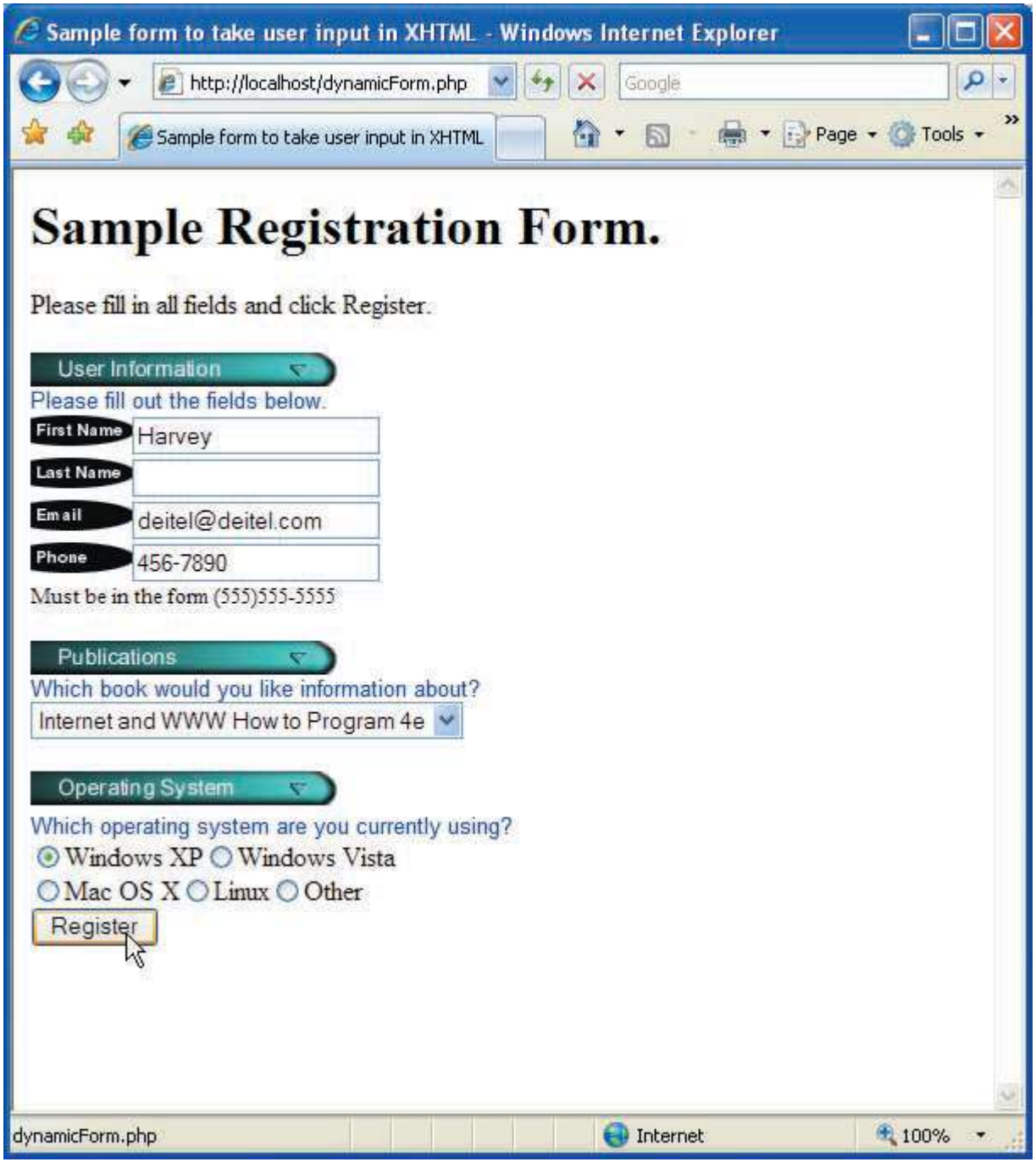
- PHP creates the superglobal array **$_COOKIE**, which contains all the cookie values indexed by their names

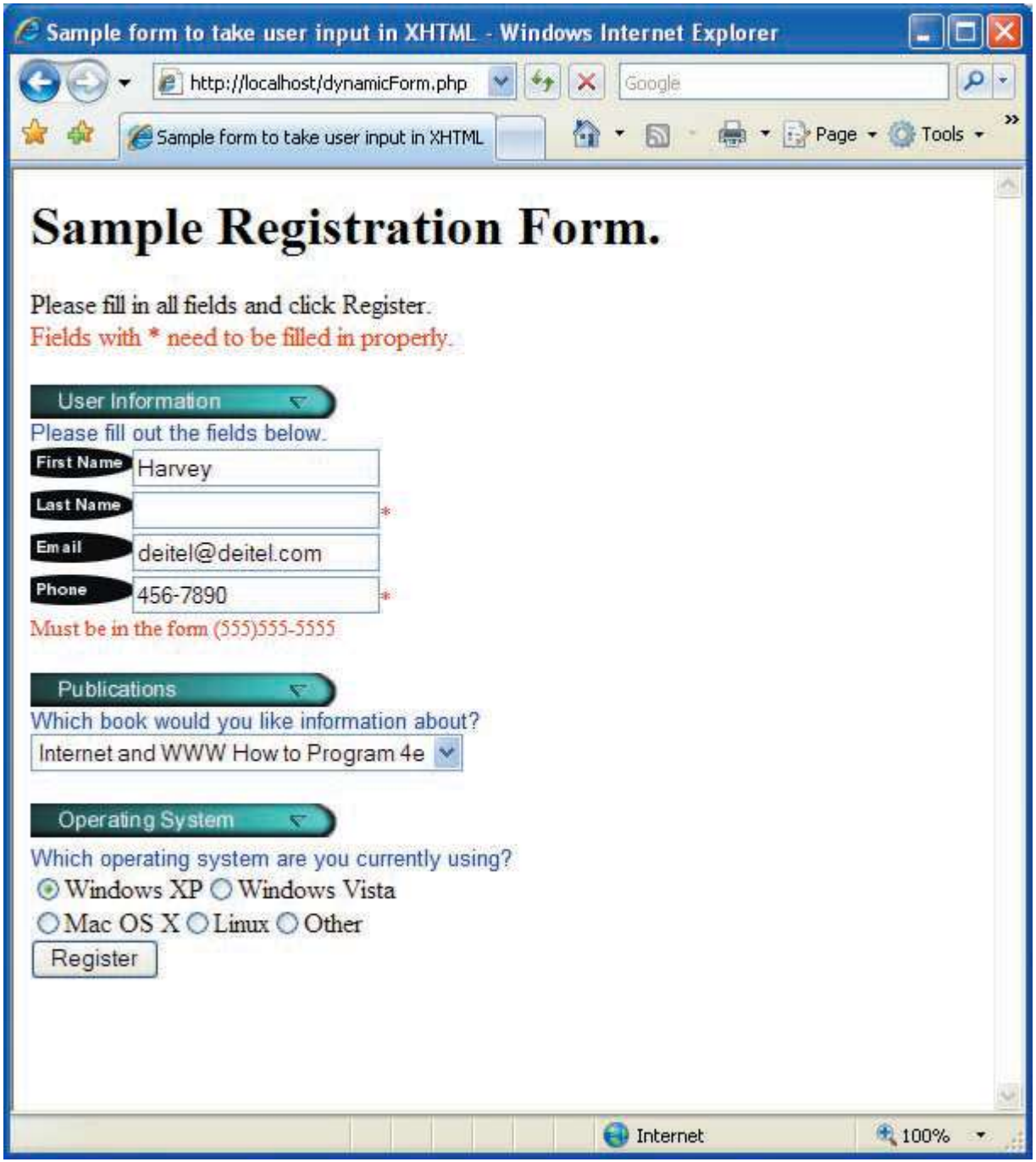
```
<body>
   <p>
      <strong>The following data is saved in a cookie on your
         computer.</strong>
   </p>
   <table>
      <?php
         // iterate through array $_COOKIE and print
         // name and value of each cookie
         foreach ( $_COOKIE as $key => $value )
            print( "<tr><td class = 'key' >$key</td>
               <td class = 'value' >$value</td></tr>" );
      ?><!-- end PHP script -->
   </table>
</body>
```

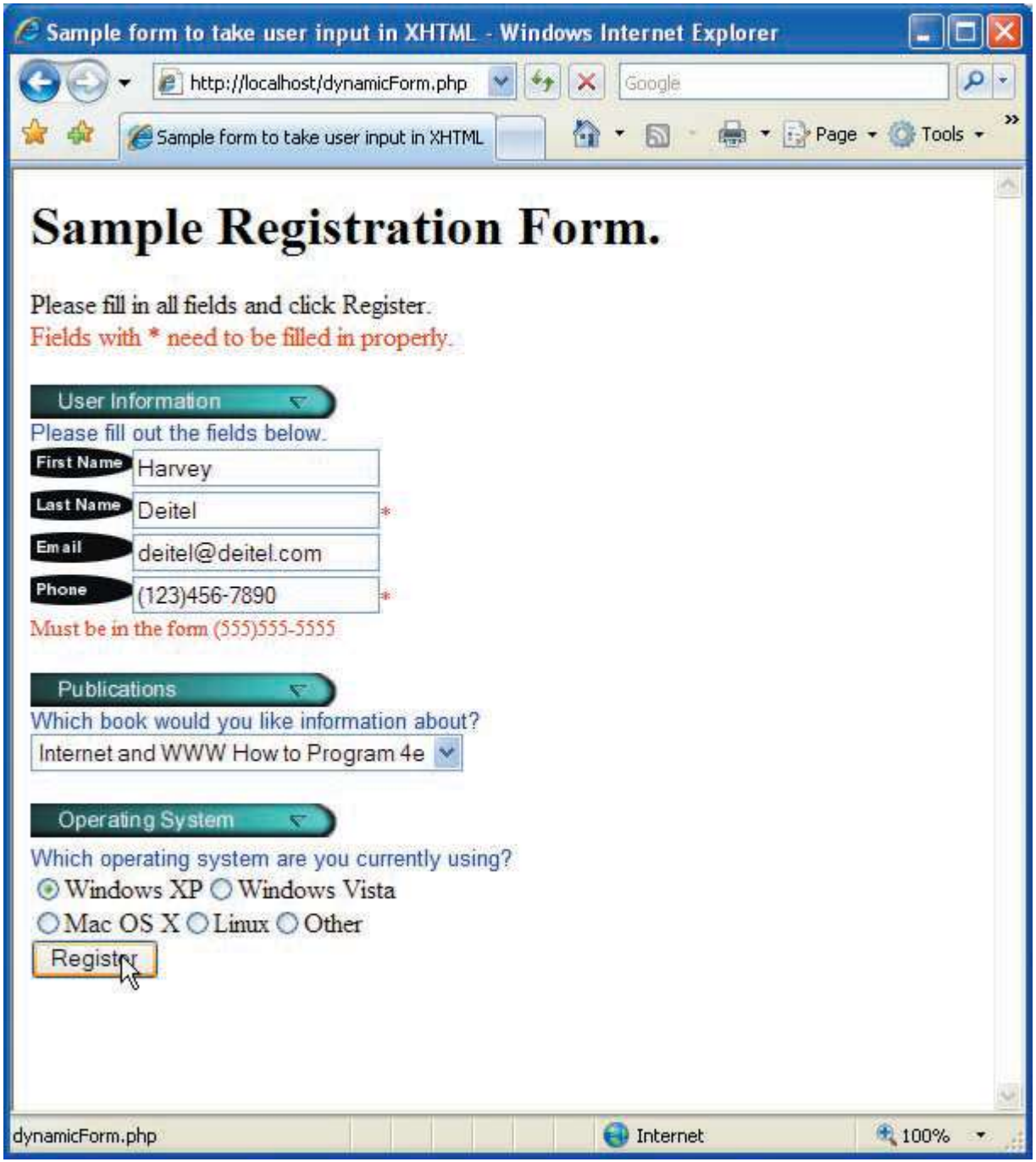

Read Cookies - Windows Internet Explorer
http://localhost/readCookies.php
Google
Read Cookies
Page
Tools
The following data is saved in a cookie on your computer.
Name
Harvey
Height
5 Feet 7 Inches
Color
Blue
Done
Internet
100%

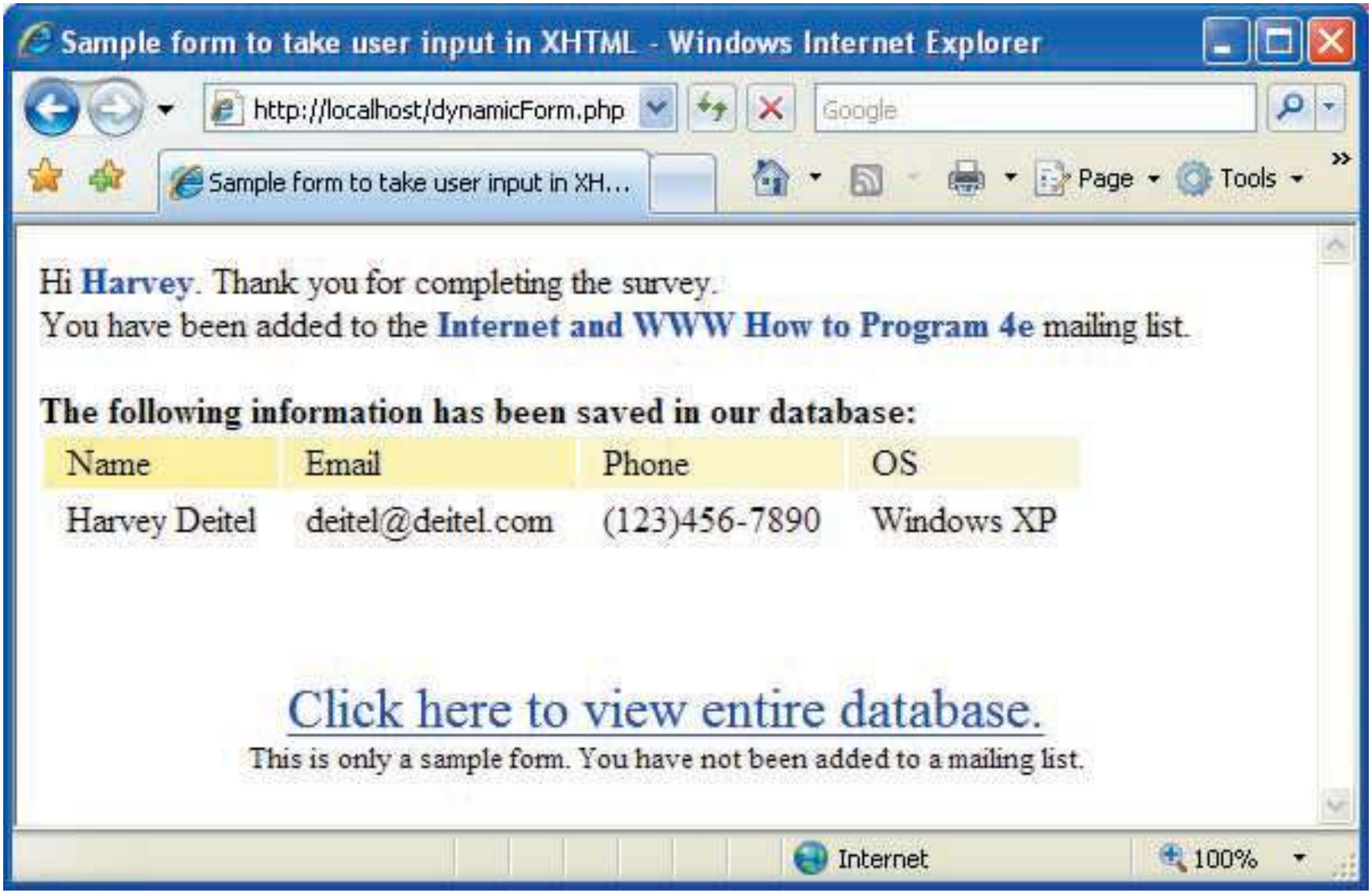
Dynamic Content

- dynamicForm.php


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost/dynamicForm.php
Google
Sample form to take user input in XHTML
Page
Tools
Sample Registration Form.
Please fill in all fields and click Register.
User Information
Please fill out the fields below.
First Name
Harvey
Last Name
Email
deitel@deitel.com
Phone
456-7890
Must be in the form (555)555-5555
Publications
Which book would you like information about?
Internet and WWW How to Program 4e
Operating System
Which operating system are you currently using?
Windows XP
Windows Vista
Mac OS X
Linux
Other
Register
dynamicForm.php
Internet
100%


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost/dynamicForm.php
Google
Sample form to take user input in XHTML
Page
Tools
Sample Registration Form.
Please fill in all fields and click Register.
Fields with * need to be filled in properly.
User Information
Please fill out the fields below.
First Name
Harvey
Last Name
*
Email
deitel@deitel.com
Phone
456-7890
*
Must be in the form (555)555-5555
Publications
Which book would you like information about?
Internet and WWW How to Program 4e
Operating System
Which operating system are you currently using?
Windows XP
Windows Vista
Mac OS X
Linux
Other
Register
Internet
100%


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost/dynamicForm.php
Google
Sample form to take user input in XHTML
Page
Tools
Sample Registration Form.
Please fill in all fields and click Register.
Fields with * need to be filled in properly.
User Information
Please fill out the fields below.
First Name
Harvey
Last Name
Deitel
*
Email
deitel@deitel.com
Phone
(123)456-7890
*
Must be in the form (555)555-5555
Publications
Which book would you like information about?
Internet and WWW How to Program 4e
Operating System
Which operating system are you currently using?
Windows XP
Windows Vista
Mac OS X
Linux
Other
Register
dynamicForm.php
Internet
100%


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost/dynamicForm.php
Google
Sample form to take user input in XH...
Page
Tools
Hi Harvey. Thank you for completing the survey.
You have been added to the Internet and WWW How to Program 4e mailing list.
The following information has been saved in our database:
Name
Email
Phone
OS
Harvey Deitel
deitel@deitel.com
(123)456-7890
Windows XP
Click here to view entire database.
This is only a sample form. You have not been added to a mailing list.
Internet
100%

```
<body>
   <?php
      extract( $_POST );
      $iserror = false;

      // array of book titles
      $booklist = array( "Internet and WWW How to Program 4e",
         "C++ How to Program 6e", "Java How to Program 7e",
         "Visual Basic 2005 How to Program 3e" );

      // array of possible operating systems
      $systemlist = array( "Windows XP", "Windows Vista",
         "Mac OS X", "Linux", "Other");

      // array of name values for the text input fields
      $inputlist = array( "fname" => "First Name",
         "lname" => "Last Name", "email" => "Email",
         "phone" => "Phone" );
```

```php
// ensure that all fields have been filled in correctly
if ( isset ( $submit ) )
{
    if ( $fname == "" )
    {
        $formerrors[ "fnameerror" ] = true;
        $iserror = true;
    } // end if

    if ( $lname == "" )
    {
        $formerrors[ "lnameerror" ] = true;
        $iserror = true;
    } // end if
```

```
if ( $email == "" )
{
    $formerrors[ "emailerror" ] = true;
    $iserror = true;
} // end if

if ( !ereg( "^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$", $phone ) )
{
    $formerrors[ "phoneerror" ] = true;
    $iserror = true;
} // end if
```

```php
if ( !$iserror )
{
   // build INSERT query
   $query = "INSERT INTO contacts " .
      "( LastName, FirstName, Email, Phone, Book, OS ) " .
      "VALUES ( '$lname', '$fname', '$email', " .
      "'" . quotemeta( $phone ) . "', '$book', '$os' )";

   // Connect to MySQL
   if ( !( $database = mysql_connect( "localhost",
      "iw3htp4", "iw3htp4" ) ) )
      die( "Could not connect to database" );

   // open MailingList database
   if ( !mysql_select_db( "MailingList", $database ) )
      die( "Could not open MailingList database" );

   // execute query in MailingList database
   if ( !( $result = mysql_query( $query, $database ) ) )
   {
      print( "Could not execute query! <br />" );
      die( mysql_error() );
   } // end if

   mysql_close( $database );
```

```
print( "<p>Hi<span class = 'prompt'>
    <strong>$fname</strong></span>.
    Thank you for completing the survey.<br />

    You have been added to the
    <span class = 'prompt'>
    <strong>$book</strong></span>
    mailing list.</p>
    <strong>The following information has been saved
    in our database:</strong><br />

    <table><tr>
    <td class = 'name'>Name </td>
    <td class = 'email'>Email</td>
    <td class = 'phone'>Phone</td>
```

```
            <td class = 'os'>OS</td>
            </tr><tr>

            <!-- print each form field's value -->
            <td>$fname $lname</td>
            <td>$email</td>
            <td>$phone</td>
            <td>$os</td>
            </tr></table>

            <br /><br /><br />
            <div><div>
            <a href = 'formDatabase.php'>
            Click here to view entire database.</a>
            </div>This is only a sample form.
            You have not been added to a mailing list.
            </div></body></html>" );
         die();
      } // end if
} // end if
```

```
print( "<h1>Sample Registration Form.</h1>
   Please fill in all fields and click Register." );

if ( $iserror )
{
   print( "<br /><span class = 'largeerror'>
      Fields with * need to be filled in properly.</span>" );
} // end if

print( "<!-- post form data to form.php -->
   <form method = 'post' action = 'dynamicForm.php'>
   <img src = 'images/user.gif' alt = 'User' /><br />
   <span class = 'prompt'>
   Please fill out the fields below.<br /> </span>
```

```
   <!-- create four text boxes for user input -->" );
foreach ( $inputlist as $inputname => $inputalt )
{
   $inputtext = $inputvalues[ $inputname ];

   print( "<img src = 'images/$inputname.gif'
      alt = '$inputalt' /><input type = 'text'
      name = '$inputname' value = '" . $$inputname . "' />" );

   if ( $formerrors[ ( $inputname )."error" ] == true )
      print( "<span class = 'error'>*</span>" );

   print( "<br />" );
} // end foreach

if ( $formerrors[ "phoneerror" ] )
   print( "<span class = 'error'>" );
```

```
else
   print("<span class = 'smalltext'>");

print( "Must be in the form (555)555-5555
   </span><br /><br />

   <img src = 'images/downloads.gif'
      alt = 'Publications' /><br />

   <span class = 'prompt'>
   Which book would you like information about?
   </span><br />

   <!-- create drop-down list containing book names -->
   <select name = 'book'>" );

foreach ( $booklist as $currbook )
{
   print( "<option" );

   if ( ( $currbook == $book ) )
      print( " selected = 'true'" );

   print( ">$currbook</option>" );
} // end foreach
```

```
print( "</select><br /><br />
   <img src = 'images/os.gif' alt = 'Operating System' />
   <br /><span class = 'prompt'>
   Which operating system are you currently using?
   <br /></span>

   <!-- create five radio buttons -->" );

$counter = 0;

foreach ( $systemlist as $currsystem )
{
   print( "<input type = 'radio' name = 'os'
      value = '$currsystem'" );

   if ( $currsystem == $os )
         print( "checked = 'checked'" );
   elseif ( !$os && $counter == 0 )
      print( "checked = 'checked'" );

   print( " />$currsystem" );

   // put a line break in list of operating systems
   if ( $counter == 1 ) print( "<br />" );
   ++$counter;
} // end foreach
```

```
      print( "<!-- create a submit button -->
         <br /><input type = 'submit' name = 'submit'
         value = 'Register' /></form></body></html>" );
?><!-- end PHP script -->
```

formDatabase.php

```
<body>
   <?php
      extract( $_POST );

      // build SELECT query
      $query = "SELECT * FROM contacts";

      // Connect to MySQL
      if ( !( $database = mysql_connect( "localhost",
         "iw3htp4", "iw3htp4" ) ) )
         die( "Could not connect to database </body></html>" );

      // open MailingList database
      if ( !mysql_select_db( "MailingList", $database ) )
         die( "Could not open MailingList database </body></html>" );

      // query MailingList database
      if ( !( $result = mysql_query( $query, $database ) ) )
      {
         print( "Could not execute query! <br />" );
         die( mysql_error() . "</body></html>" );
      } // end if
   ?><!-- end PHP script -->
```

```
<h3>Mailing List Contacts</h3>
<table>
    <tr>
        <td>ID</td>
        <td>Last Name</td>
        <td>First Name</td>
        <td>E-mail Address</td>
        <td>Phone Number</td>
        <td>Book</td>
        <td>Operating System</td>
    </tr>
    <?php
```

```
         // fetch each record in result set
         for ( $counter = 0; $row = mysql_fetch_row( $result );
            $counter++ )
         {
            // build table to display results
            print( "<tr>" );

            foreach ( $row as $key => $value )
               print( "<td>$value</td>" );

            print( "</tr>" );
         } // end for

         mysql_close( $database );
      ?><!-- end PHP script -->
   </table>
</body>
```

Reference

- “Internet & Wrold Wide Web – How to Program” , Forth Edition
  - P.J. Deitel & H.M. Deitel

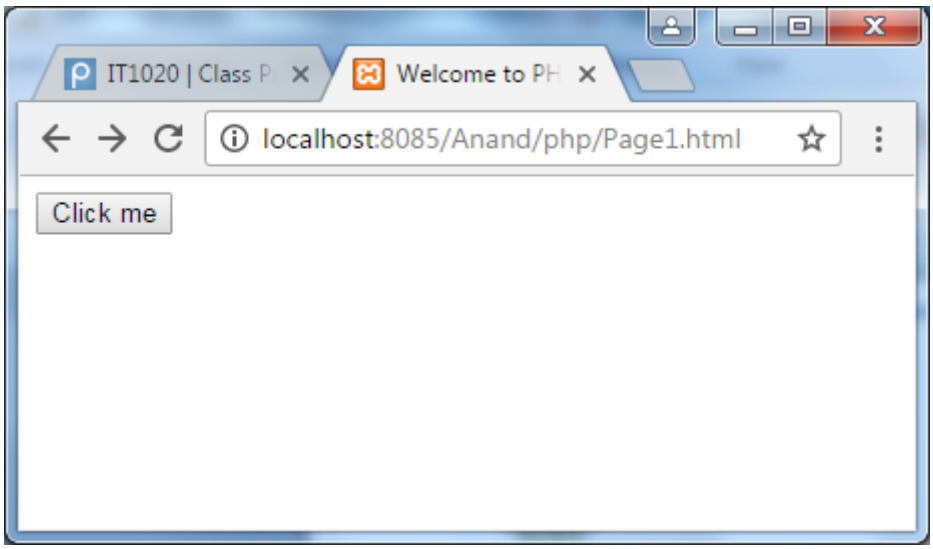
PHP Sample Programs

Example 1: Simple PHP program

- **Using method and action attributes of form element**
- **Using variables and echo statement**

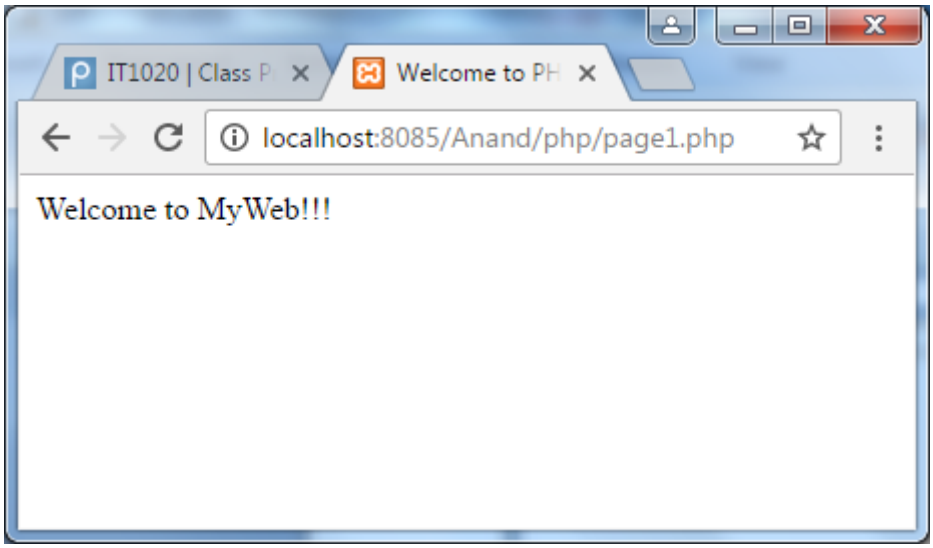
page1.html

```
<html>
<head>
   <title>Welcome to PHP</title>
</head>
<body>
<form method="post" action="page1.php">
        <input type=submit value="Click me">
</form>
</body>
</html>
```



Page1.php

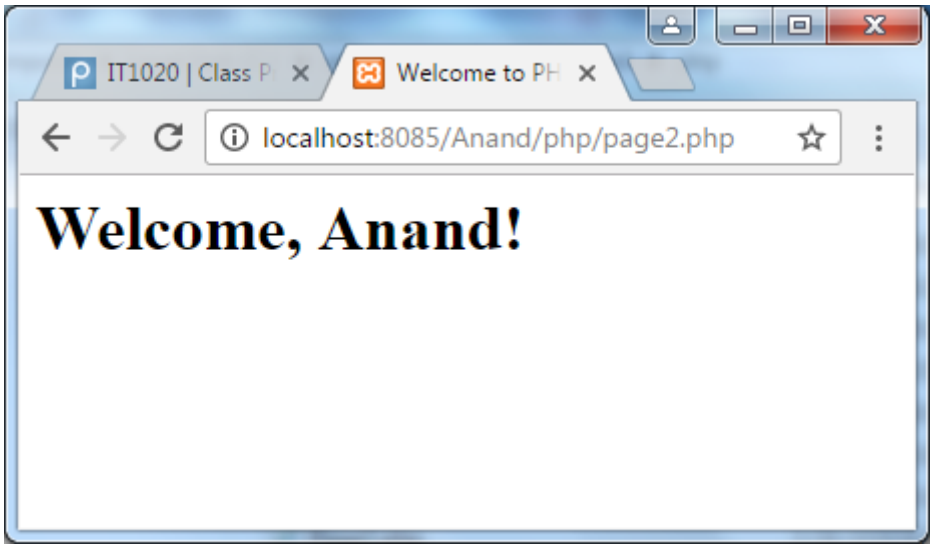
```
<html>
<head>
   <title>Welcome to PHP</title>
        <?php $msg="Welcome to MyWeb!!!"; ?>
</head>
<body>
        <?php echo "$msg"; ?>
</body>
</html>
```



Example 2: Using print function

page2.php

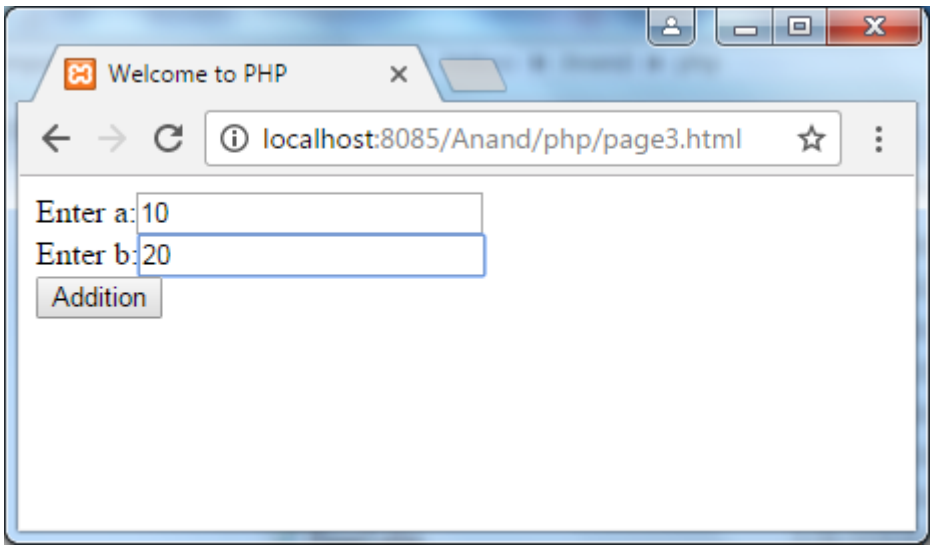
```
<html>
<head>
   <title>Welcome to PHP</title>
   <?php $name="Anand"; ?>
</head>
<body>
        <H1> <?php print("Welcome, $name!");?> </H1>
</body>
</html>
```



Example 3: Add two numbers using $_POST()

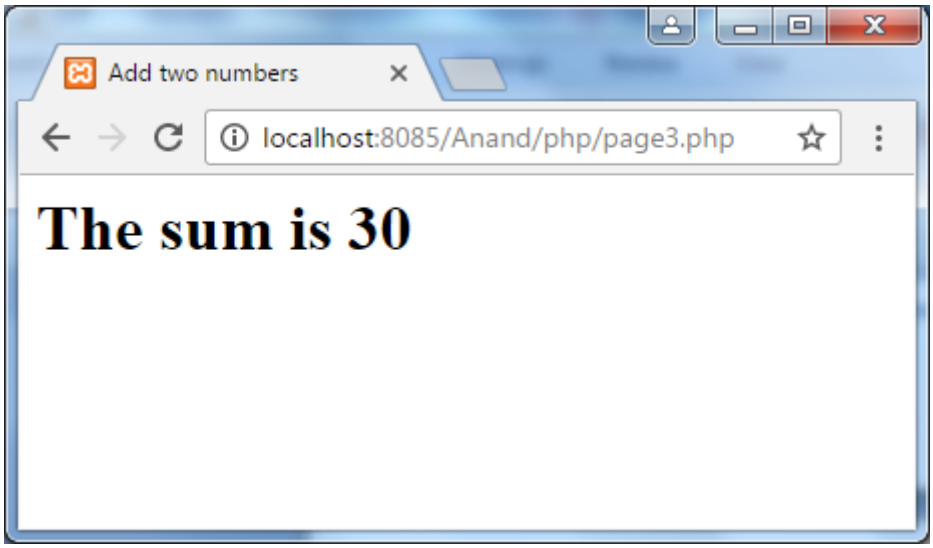
page3.html

```
<html>
<head>
   <title>Welcome to PHP</title>
        <?php $msg="Welcome to PHP"; ?>
</head>
<body>
<form method="post" action="page3.php">
        Enter a:<input type=text name=a><br>
        Enter b:<input type=text name=b><br>
        <input type=submit value="Addition">
</form>
</body>
</html>
```



page3.php

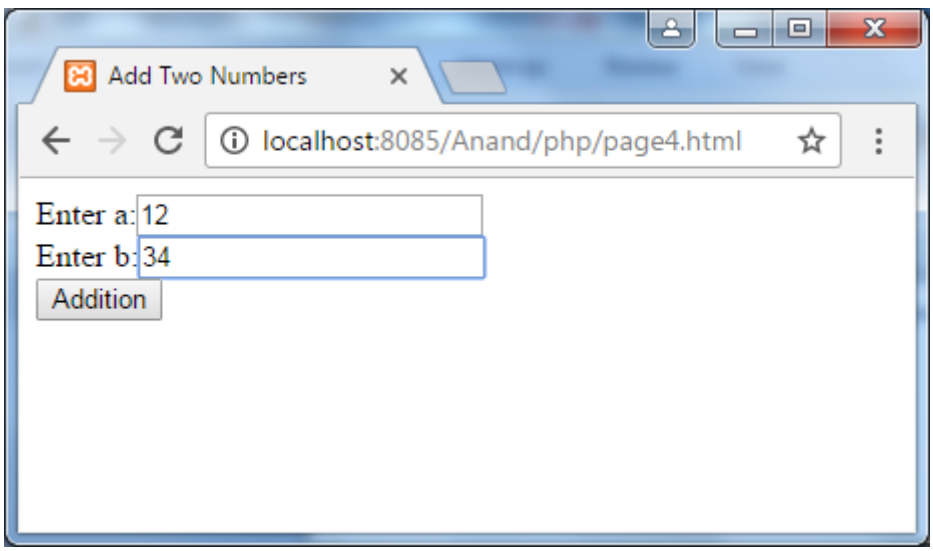
```
<html>
<head>
   <title>Add two numbers</title>
</head>
<body>
<?php
        $a=$_POST['a'];
        $b=$_POST['b'];
        $c=$a+$b;
        print("<h1>The sum is $c</h1>");
?>
</body>
</html>
```



Example 4: Add two numbers using extract()

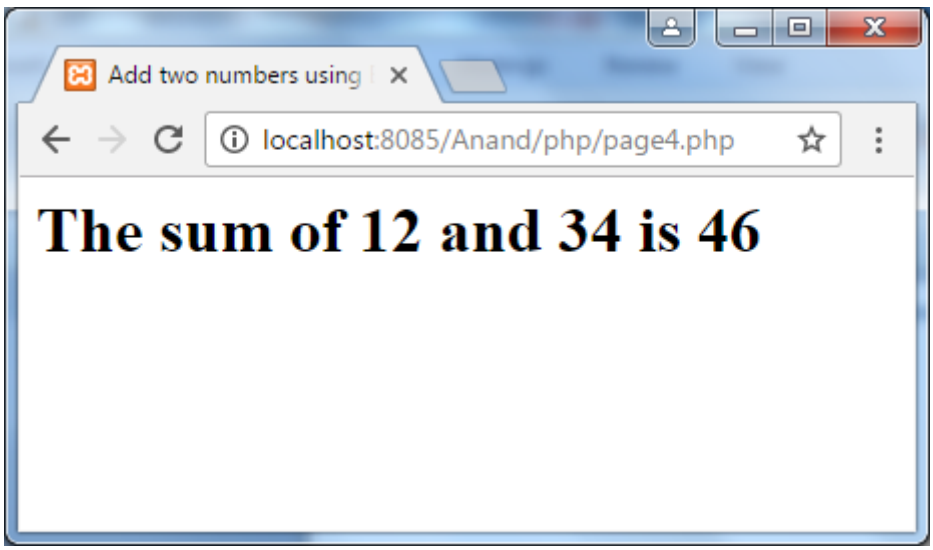
Page4.html

```
<html>
<head>
   <title>Add Two Numbers</title>
</head>
<body>
<form method="post" action="page4.php">
        Enter a:<input type=text name=a><br>
        Enter b:<input type=text name=b><br>
        <input type=submit value="Addition">
</form>
</body>
</html>
```



Page4.php

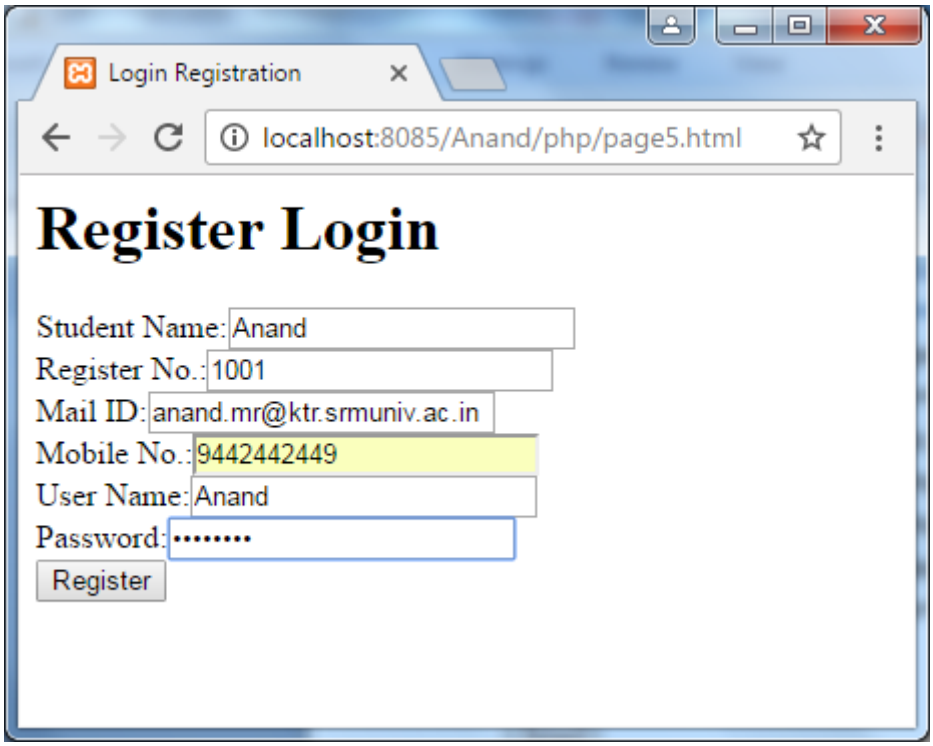
```
<html>
<head>
   <title>Add two numbers using Extract</title>
</head>
<body>
<?php
        extract($_POST);
        $c=$a+$b;
        print("<h1>The sum of $a and $b is $c</h1>");
?>
</body>
</html>
```



Example 5: Creating user registration page using PHP code in HTML

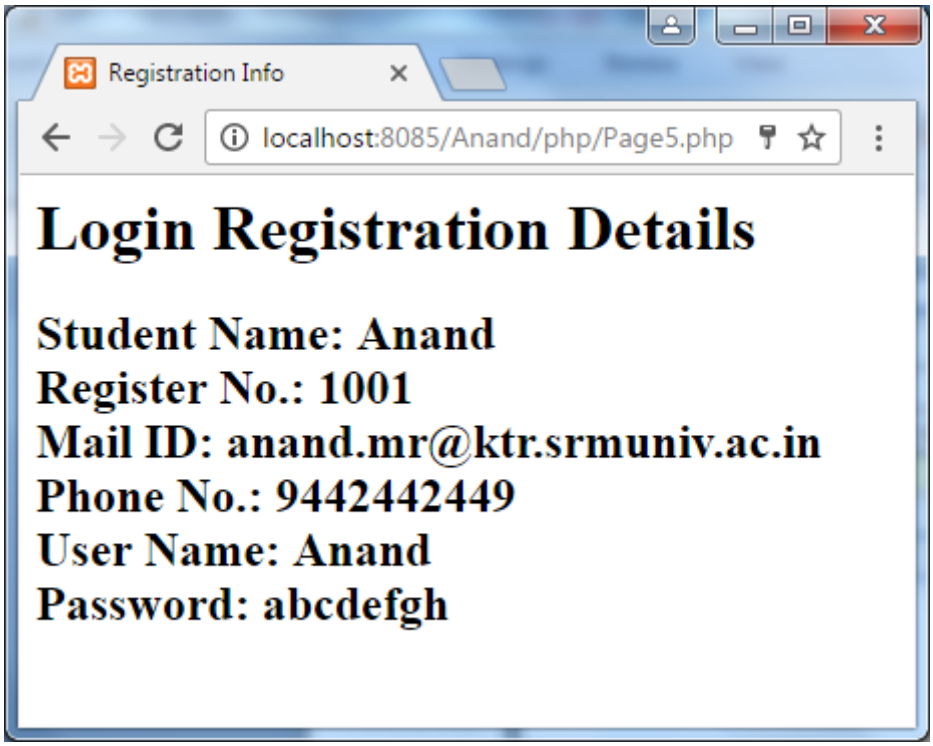
page5.html

```
<html>
<head>   <title>Login Registration</title>   </head>
<body>
<form method="post" action="Page5.php">
        <H1>Register Login</h1>
        Student Name:<input type=text name=sname><br>
        Register No.:<input type=text name=regno><br>
        Mail ID:<input type=text name=mailid><br>
        Mobile No.:<input type=text name=mobile><br>
        User Name:<input type=text name=uname><br>
        Password:<input type=password name=pwd><br>
        <input type=submit value="Register">
</form>
</body>
</html>
```



page5.php

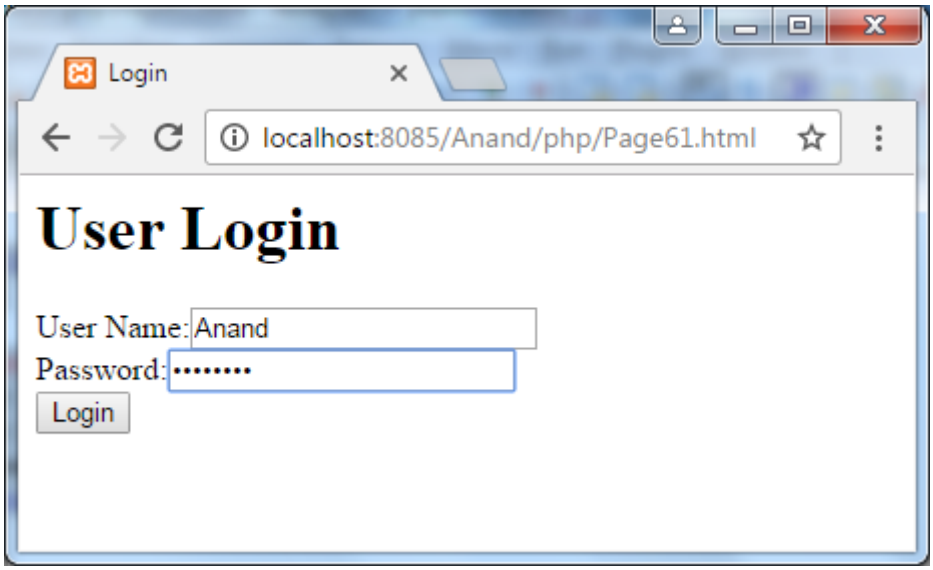
```
<html>
<head>
   <title>Registration Info</title>
</head>
<body>
<form method="post">
        <?php
                extract($_POST);
        ?>
        <h1>Login Registration Details</h1>
        <h2>
        Student Name: <?php print($sname) ?><br>
        Register No.: <?php print($regno) ?></br>
        Mail ID: <?php print($mailid) ?><br>
        Phone No.: <?php print($mobile) ?><br>
        User Name: <?php print($uname) ?><br>
        Password: <?php print($pwd) ?><br>
        </h2>
</form>
</body>
</html>
```



Example 6: Redirecting a page using header() function

Page61.html

```
<html>
<head>
   <title>Login</title>
</head>
<body>
<form method="post" action="Page6.php">
<H1>User Login</h1>
          User Name:<input type=text name=uname><br>
          Password:<input type=password name=pwd><br>
          <input type=submit value="Login">
</form>
</body>
</html>
```

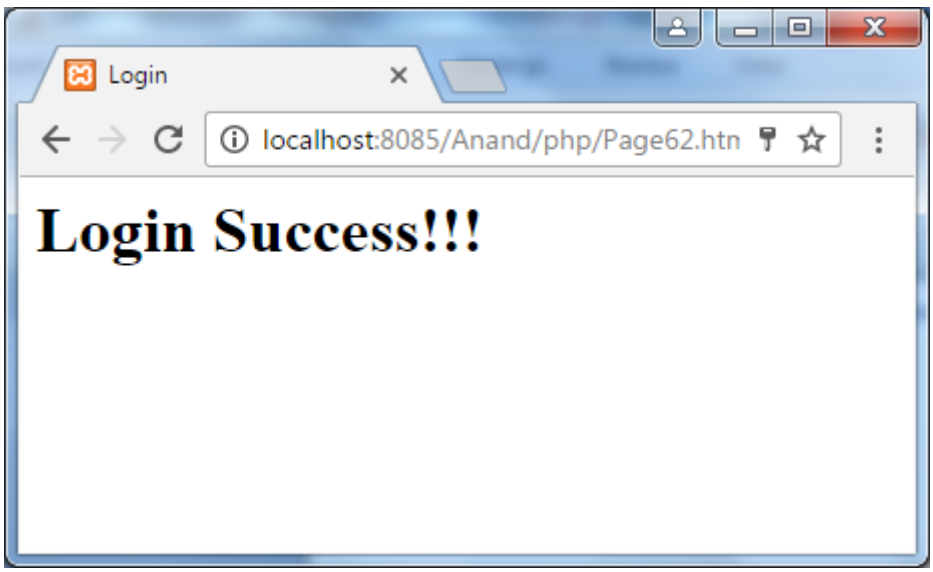


Page6.php

```
<?php
if (($_POST['uname']=="Anand") && ($_POST['pwd']="anand123")){
        header("location:Page62.html");
}
?>
```

Page62.html

```
<html>
<head>
   <title>Login</title>
</head>
<body>
        <H1>Login Success!!!</h1>
</form>
</body>
</html>
```



DATABASE PROGRAMS

Creating database in *phpmyadmin*

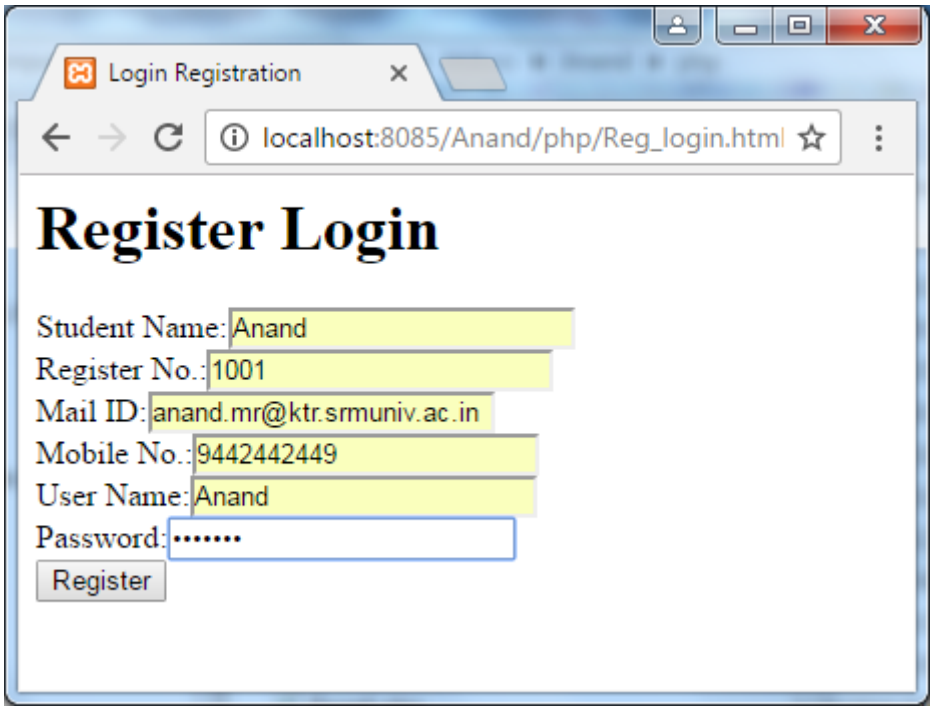
Database Name: **studb**
Table Name: **login**

Field Name	Data Type
sname	Varchar(50)
regno	Varchar(20)
mailid	Varchar(100)
mob	Varchar(15)
uame	Varchar(50)
pwd	Varchar(50)

Example 7: Creating user registration page with database

Reg_login.html

```
<html>
<head>
   <title>Login Registration</title>
</head>
<body>
<form method="post" action="Reg_Save.php">
<H1>Register Login</h1>
        Student Name:<input type=text name=sname><br>
        Register No.:<input type=text name=regno><br>
        Mail ID:<input type=text name=mailid><br>
        Mobile No.:<input type=text name=mobile><br>
        User Name:<input type=text name=uname><br>
        Password:<input type=password name=pwd><br>
        <input type=submit value="Register">
</form>
</body>
</html>
```

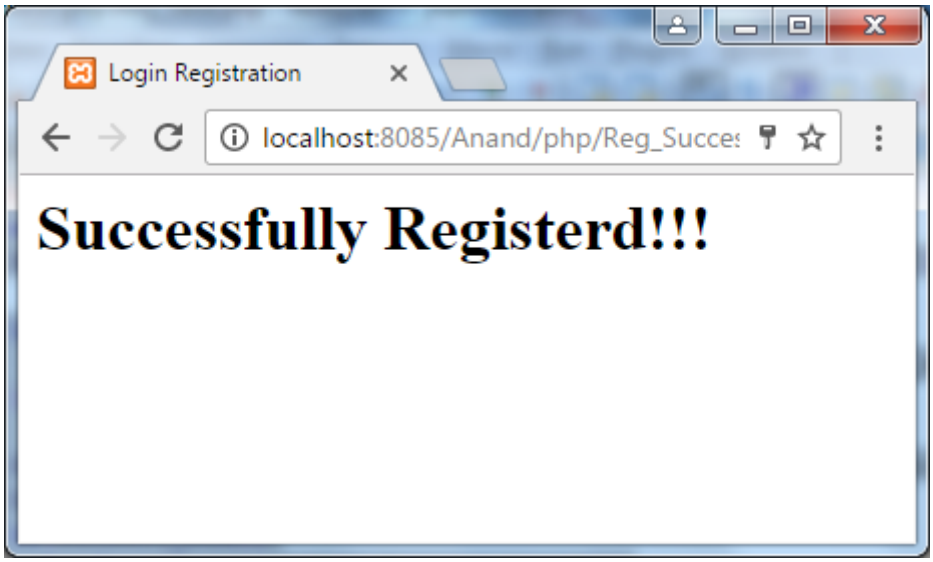


Reg_Save.php

```
<?php
        extract($_POST);
        $mysqli = new mysqli('localhost', 'root','server','studb');

        if($mysqli->connect_errno > 0){
        die('Unable to connect to database [' . $db->connect_error . ']');
        }

        $query = "INSERT INTO login VALUES
                ('$sname','$regno','$mailid','$mobile','$uname','$pwd')";
        $insert_row = $mysqli->query($query);
        if($insert_row){
          header("location:Reg_Success.html");
        }
        else{
        die('Error : (. $mysqli->errno .') '. $mysqli->error);
        }
?>
```

Reg_Success.html

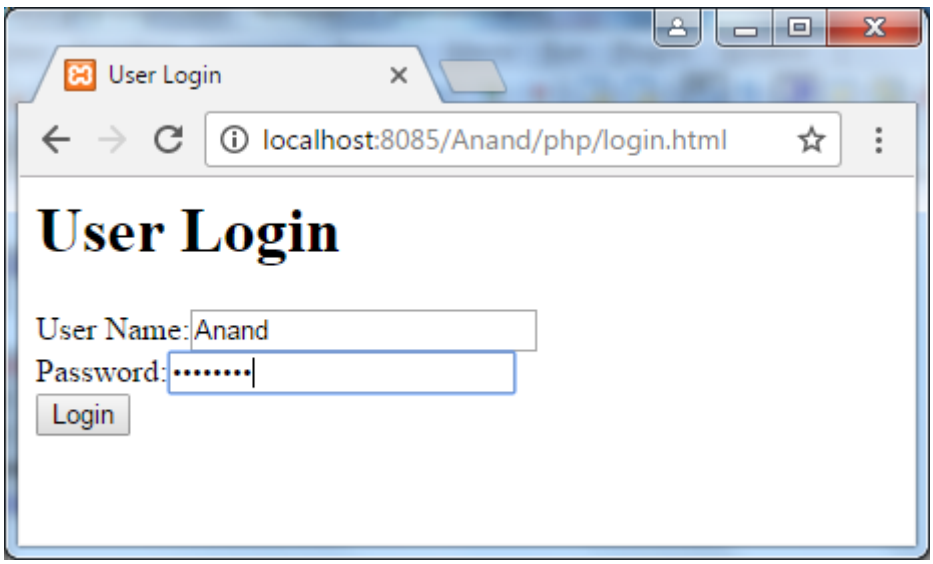
```
<html>
<head>
   <title>Login Registration</title>
</head>
<body>
        <H1>Successfully Registerd!!!</h1>
</form>
</body>
</html>
```



Example 8: Creating user login page in PHP with database

Login.html

```
<html>
<head>
   <title>User Login</title>
</head>
<body>
<form method="post" action="Login_Check.php">
<H1>User Login</h1>
        User Name:<input type=text name=uname><br>
        Password:<input type=password name=pwd><br>
        <input type=submit value="Login">
</form>
</body>
</html>
```

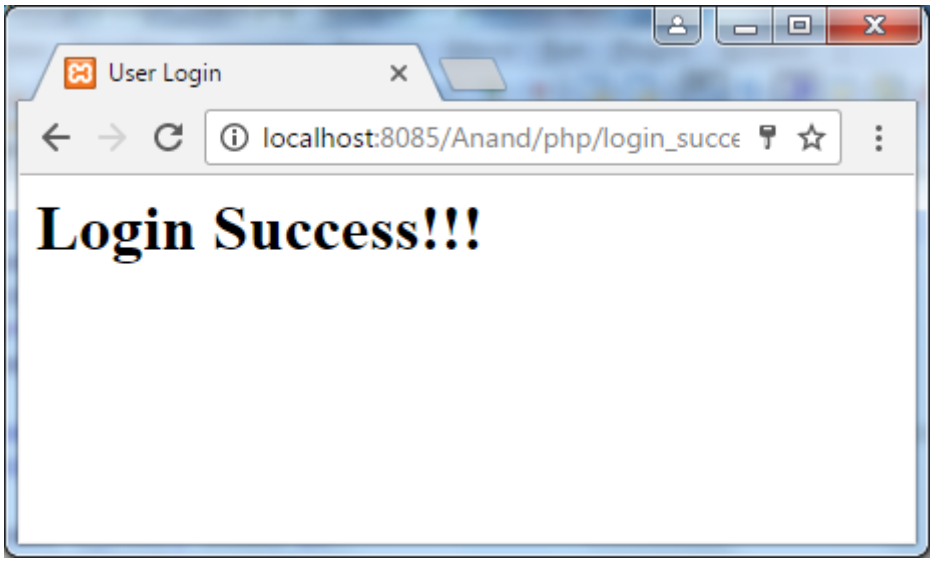


Login_Check.php

```
<?php
$host="localhost";
$username="root";
$password="server";
$db_name="studb";
$tbl_name="Login";

mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

$myusername=$_POST['uname'];
$mypassword=$_POST['pwd'];

$sql="SELECT * FROM $tbl_name WHERE uname='$myusername' and
pwd='$mypassword'";
$result=mysql_query($sql);

$count=mysql_num_rows($result);

if($count==1){
        header("location:login_success.html");
}
else {
        echo "<h2>Login Failed!!!</h2>";
}
?>
```

Login_Success.html

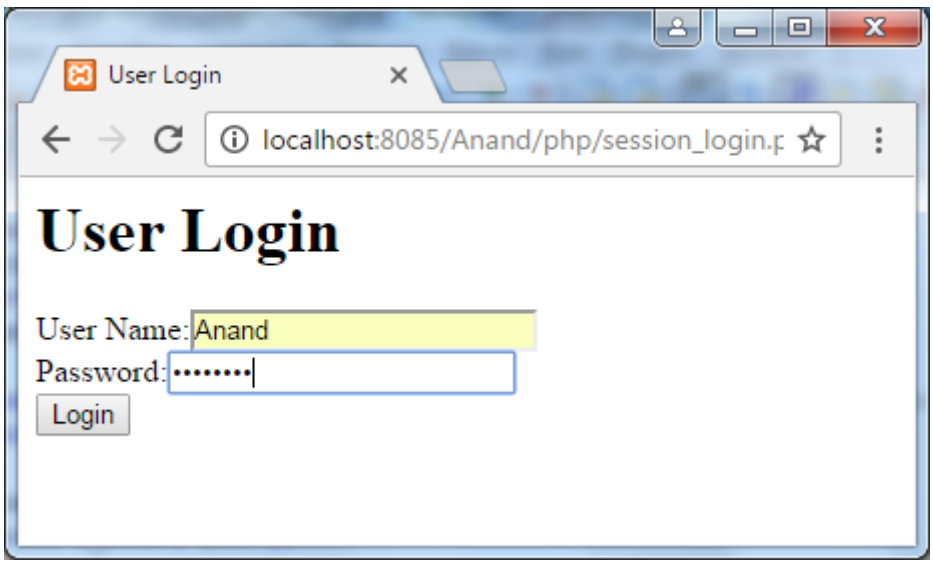
```
<html>
<head>
   <title>User Login</title>
</head>
<body>
        <H1>Login Success!!!</h1>
</form>
</body>
</html>
```



Example 9: Creating user login page using session

Session_Login.php

```
<html>
<head>
   <title>User Login</title>
</head>
<body>
<form method="post" action="Session_Login_Check.php">
        <H1>User Login</h1>
        User Name:<input type="text" name="uname"><br>
        Password:<input type="password" name="pwd"><br>
        <input type="submit" value="Login">
</form>
</body>
</html>
```



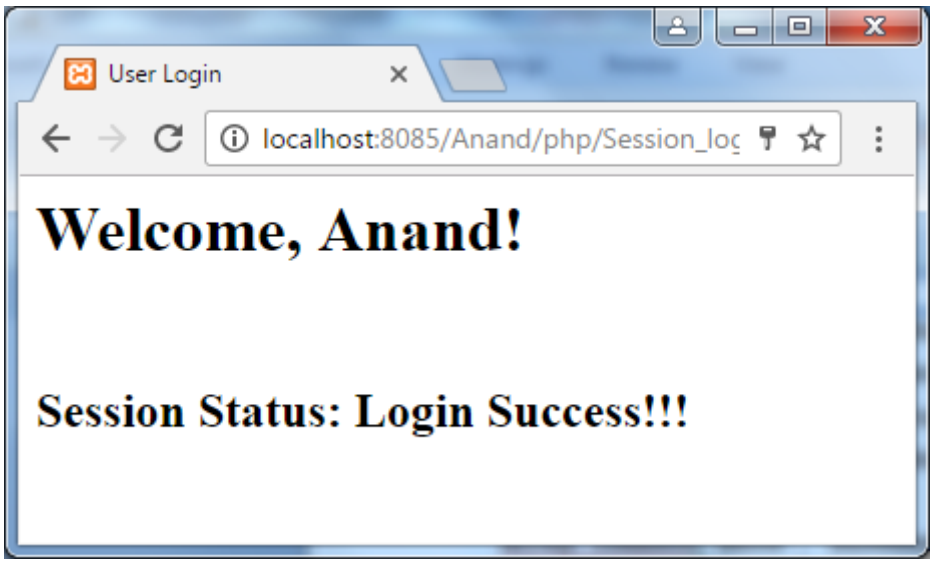
Session_Login_Check.php

```
<?php
$host="localhost";
$username="root";
$password="server";
$db_name="studb";
$tbl_name="Login";

mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

$myusername=$_POST['uname'];
$mypassword=$_POST['pwd'];

$sql="SELECT * FROM $tbl_name WHERE uname='$myusername' and
pwd='$mypassword'";
$result=mysql_query($sql);

$count=mysql_num_rows($result);

if($count==1){
        $row=mysql_fetch_row($result);
        $sname=$row[0];
        session_start();
        $_SESSION['SName']=$sname;
        $_SESSION['Status']="Session Status: Login Success!!!";

        header("location:Session_login_success.php");
}
else {
        session_start();
        if (isset($_SESSION['SName'])){
                unset($_SESSION['SName']);
        }
        $_SESSION['Status']="Session Expired!!!";
        echo "<h2>Login Failed!!!</h2>";
}
?>
```

Session_Login_Success.php

```
<html>
<head>
   <title>User Login</title>
</head>
<body>
```

```
        <?php
                session_start();
                if (isset($_SESSION['SName'])){
                        echo "<h1>Welcome, ".$_SESSION['SName']."!</h1><br>";
                        echo "<h2>".$_SESSION['Status']."</h2>";
                }
                else {
                        echo "<h2>".$_SESSION['Status']."</h2>";
                        print("<h4><br><br><a href='Session_Login.php'>
                                                        Login again!</a></h4>");
                        //header("location:Session_login.php");
                }
        ?>
        </h1>
</body>
</html>
```

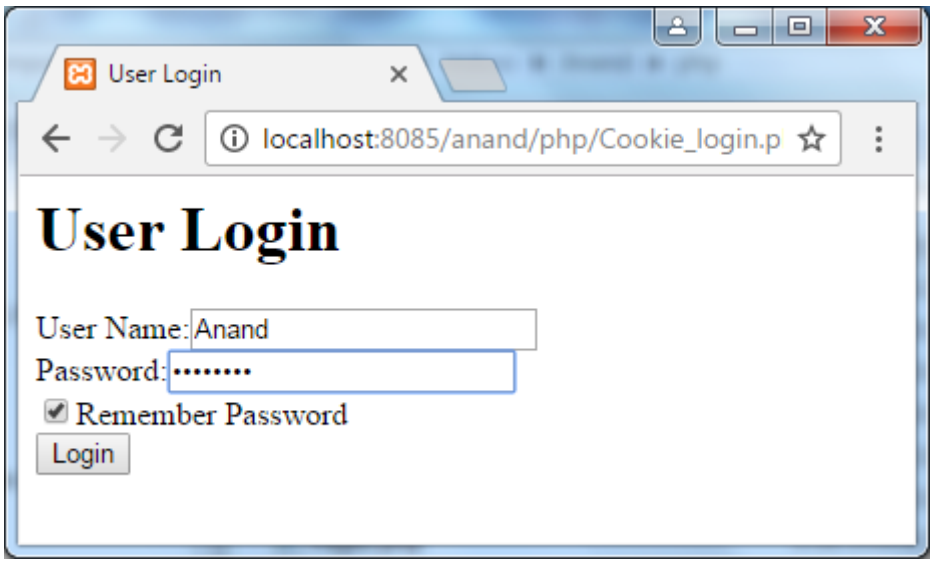


Example 10: Creating user login page using cookies

Cookie_Login.php

```
<html>
<head>
   <title>User Login</title>
</head>
<body>
<form method="post" action="Cookie_Login_Check.php">
  <H1>User Login</h1>
  User Name:<input type="text" name="uname" value=<?php echo Cookie_val(1) ?> > <br>
  Password:<input type="password" name="pwd" value=<?php echo Cookie_val(2) ?> >
  <br>
  <input type="checkbox" name="remuser">Remember Password<br>
  <input type="submit" value="Login">
</form>
```

```
</body>
	<?php
		function Cookie_val($ch){
		if ($ch==1)
		{
			if (isset($_COOKIE['UName']))
			{
				return $_COOKIE['UName'];
			}
			else
				return;
		}
		else
		{
			if (isset($_COOKIE['Pwd']))
			{
				return $_COOKIE['Pwd'];
			}
			else
				return;
		}
		}
	?>
</html>
```



Cookie_Login_Check.php

```
<?php
$host="localhost";
$username="root";
$password="server";
$db_name="studb";
$tbl_name="Login";
```

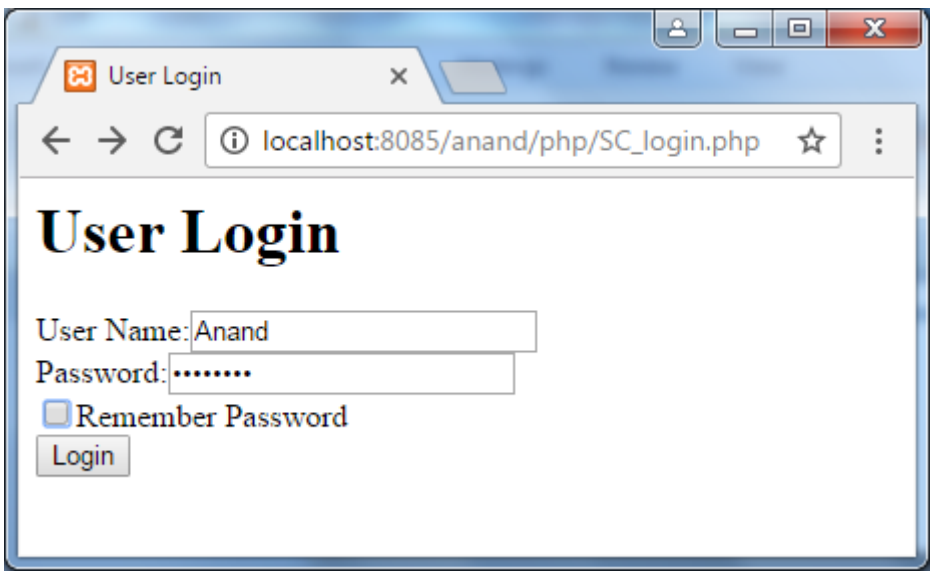
```
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

$myusername=$_POST['uname'];
$mypassword=$_POST['pwd'];

$sql="SELECT * FROM $tbl_name WHERE uname='$myusername' and
pwd='$mypassword'";
$result=mysql_query($sql);

$count=mysql_num_rows($result);

if($count==1){
        $row=mysql_fetch_row($result);
        $sname=$row[0];
        print("<h1>Welcome, $sname!<br></h1>");
        echo "<h2>Login Success!!!</h2>";

        if (isset($_POST['remuser'])){
                setcookie("UName",$myusername,time()+60*60*24);
                setcookie("Pwd",$mypassword,time()+60*60*24);
        }
        else
        {
                setcookie("UName","",time()-5);
                setcookie("Pwd","",time()-5);
        }

//      header("location:login_success.html");
}
else {
        echo "<h2>Login Failed!!!</h2>";
        //header("location:login_failed.php");
}
?>
```

Example 11: Creating user login page using session and cookies

SC_Login.php

```
<html>
<head>
   <title>User Login</title>
</head>
<body>
```

```
<form method="post" action="SC_Login_Check.php">
  <H1>User Login</h1>
  User Name:<input type="text" name="uname" value=<?php echo Cookie_val(1) ?> ><br>
  Password:<input type="password" name="pwd" value=<?php echo Cookie_val(2) ?> >
            <br>
  <input type="checkbox" name="remuser">Remember Password<br>
  <input type="submit" value="Login">
</form>
</body>
      <?php
            function Cookie_val($ch){
            if ($ch==1)
            {
                  if (isset($_COOKIE['UName']))
                  {
                        return $_COOKIE['UName'];
                  }
                  else
                        return;
            }
            else
            {
                  if (isset($_COOKIE['Pwd']))
                  {
                        return $_COOKIE['Pwd'];
                  }
                  else
                        return;
            }
            }
      ?>
</html>
```

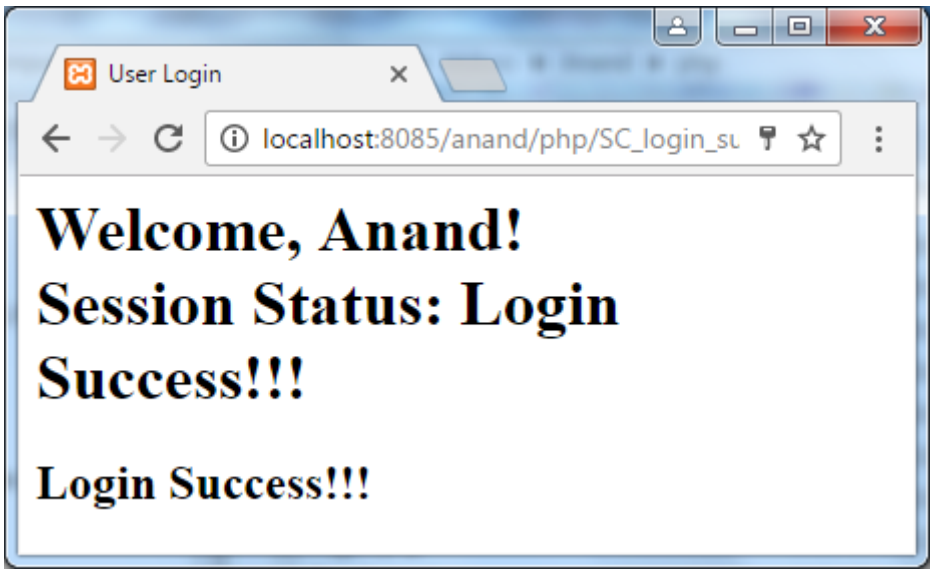


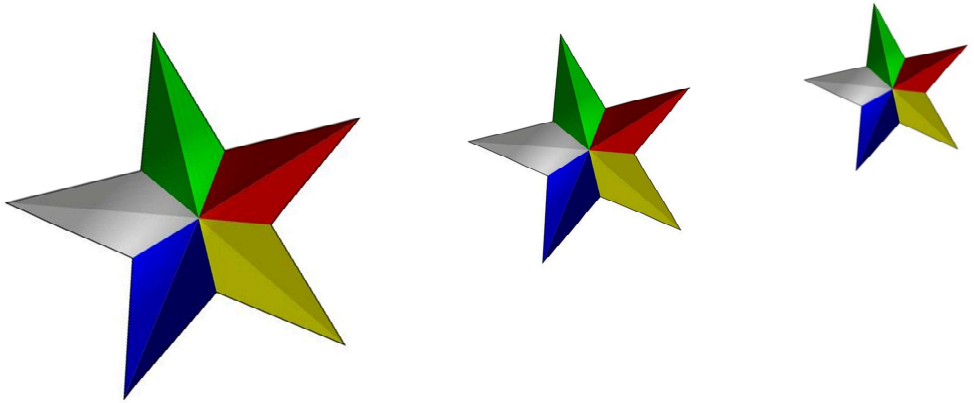
SC_Login_Check.php

```
<?php
$host="localhost";
$username="root";
$password="server";
$db_name="studb";
$tbl_name="Login";

mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

$myusername=$_POST['uname'];
$mypassword=$_POST['pwd'];

$sql="SELECT * FROM $tbl_name WHERE uname='$myusername' and
pwd='$mypassword'";
$result=mysql_query($sql);

$count=mysql_num_rows($result);

if($count==1){
	$row=mysql_fetch_row($result);
	$sname=$row[0];
	session_start();
	$_SESSION['SName']=$sname;
	$_SESSION['Status']="Session Status: Login Success!!!";

	if (isset($_POST['remuser'])){
		setcookie("UName",$myusername,time()+60*60*24);
		setcookie("Pwd",$mypassword,time()+60*60*24);
	}
	else
	{
		setcookie("UName","",time()-5);
		setcookie("Pwd","",time()-5);
	}
	header("location:SC_login_success.php");
}
else {
	unset($_SESSION['SName']);
	unset($_SESSION['Status']);
	echo "<h2>Login Failed!!!</h2>";
	header("location:SC_login.php");
}

?>
```

SC_Login_Success.php

```
<html>
<head>
   <title>User Login</title>
</head>
<body>
        <h1>
        <?php
                session_start();
                if (isset($_SESSION['SName'])){
                        echo "Welcome, ".$_SESSION['SName']."!<br>";
                        echo $_SESSION['Status'];
                }
                else {
                        header("location:SC_login.php");
                }
        ?>
        </h1>
        <h2>Login Success!!!</h2>
</form>
</body>
</html>
```





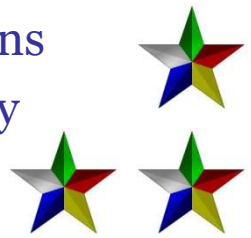
Working with XML

Dr. M. Anand

Assistant Professor (Sr. G)

Dept. of Networking and Communications

SRM Institute of Science and Technology



XML

- XML
  - Introduction
  - Syntax
  - DTD
  - XMLSchema
- Displaying XML Documents
  - CSS
  - XSLT
- XML Parsers
  - SAX
  - DOM
- XML Applications
  - RSS/Atom
  - SOAP

What is XML?

- Extensible Markup Language (XML)
- ***"XML is a cross-platform, software and hardware independent tool for transmitting information"***
- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive

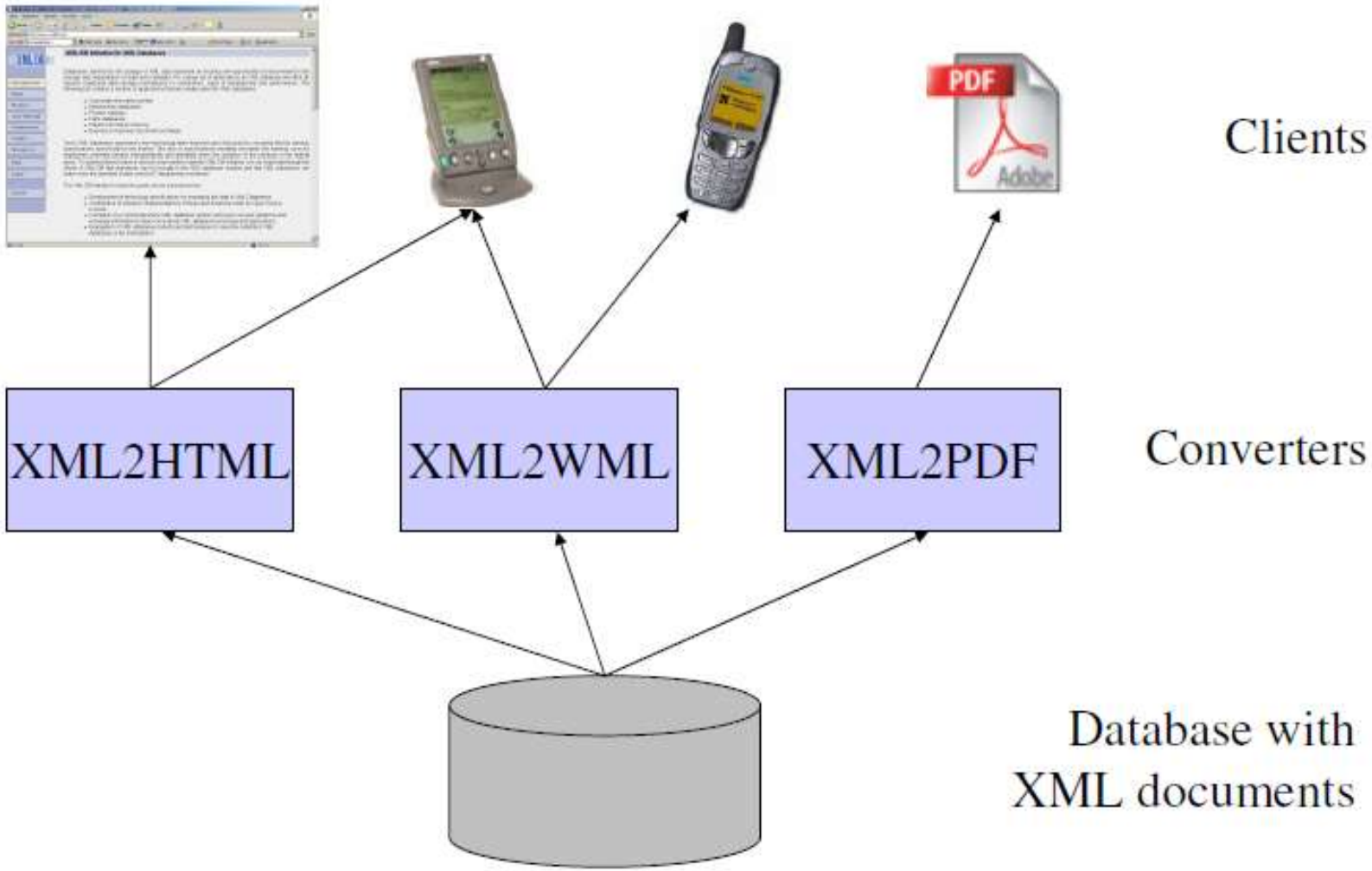
What is XML?

- XML describes data as a combination of markup language, like HTML
- The flat, relational, or hierarchical data can be represented easily in XML
- It is represented in a text-only format
- Platform independent
- Recommended by W3C

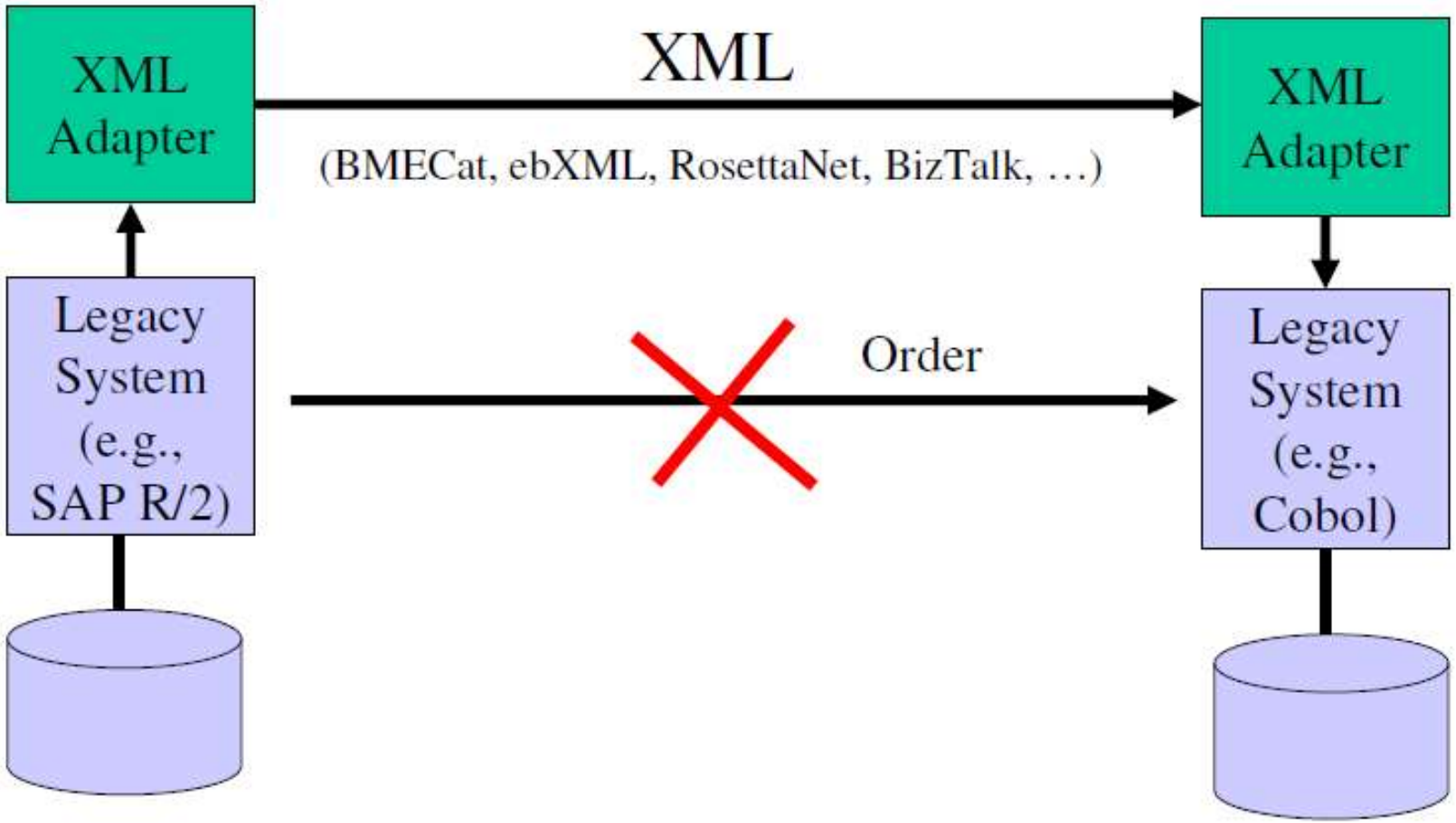
Role of XML

- Role of XML
  - Configuration files of an application or Web pages are deployed in XML
  - Objects are stored across the internet by serializing them into XML
  - Web Services intercommunication is based on XML

App. Scenario 1:Content Management



App. Scenario 2: Data Exchange



Markup Language

- A markup language is the set of rules that declares and defines the elements.
- It also provides a description of document layout and logical structure.

Markup Language

- Three types of markup
  - **Stylistic:** How a document is presented
    - e.g., the HTML tags <I> for italics, <B> for bold, and <U> for underline
  - **Structural:** How the document is to be structured
    - e.g., the HTML tags <P> for paragraph, <SPAN> for creating ad hoc styles in a document, and <DIV> for grouping structures
  - **Semantic:** Tells about the content of the data
    - e.g., the HTML tags <TITLE> for page title, <HEAD> for page header information, and <SCRIPT>to indicate a JavaScript in a page

Simple XML Document

```
<?xml version="1.0" encoding="UTF-8" ?>
<greeting>Hello, world!</greeting>
```

XML Display

- List of components necessary to view XML documents
  - Document Type Definitions (DTD)
  - XML aware browsers
  - CSS (Cascading Style Sheets) and/or XSL(Extensible Stylesheet Language (XSL)
  - XML Parsers
  - Namespaces

Comparing XML with HTML

- XML is designed to carry data
- HTML focuses on display
- XML tags are not predefined
- XML tags are case sensitive
- XML is not a replacement but a complement to HTML

XML Syntax

- XML Declaration
- Opening and Closing Tags
- Root Element
- Child Elements
- Nested Elements
- XML Attribute Values
- White space is preserved
- Comments in XML

XML Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<card>
  <name>Sachin Tendlya</name>
  <title>CEO, Master Blaster Inc.</title>
  <email>sachin.tendyla@masterblaster.com</email>
  <phone>(022) 333-4567</phone>
</card>
```

XML Attributes

- Attributes provide information about data
- Attributes are not relevant to the reader
- Only metadata(data about data should be stored as attributes)

```
<?xml version="1.0" encoding="UTF-8"?>
<card>
 <name>Sachin Tendlya</name>
 <title>CEO, Master Blaster Inc.</title>
 <email>sachin.tendyla@masterblaster.com</email>
 <phone number=“Residence”>(022) 333-4567</phone>
</card>
```

XML Syntax Rules

- All XML Elements Must Have a Closing Tag
- XML Tags are Case Sensitive
- XML Elements Must be Properly Nested
- XML Documents Must Have a Root Element
- XML Attribute Values Must be Quoted
- Entity References
- Comments in XML

XML Validation

- XML with correct syntax is Well formed XML
- DTD (Document Type Definition) validation
  - XML validated against DTD is Valid XML
  - DTD specifies how the XML data is structured
  - DTD can be Internal or External
- Validation with XMLSchema

Internal DTD

```
<?xml version="1.0"?>
<!DOCTYPE E-mail[
  <!ELEMENT E-mail (To,From,subject,Body)>
  <!ELEMENT To (#PCDATA)>
  <!ELEMENT From (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
  <!ELEMENT Body (#PCDATA)>
]>
<Email>
  <To>Rohan</To>
  <From>Amit</From>
  <Subject>Movie </Subject>
  <Body>Be ready for the Movie. I will come at 7PM
  </Body>
</Email>
```

External DTD

- <!DOCTYPE root-element SYSTEM "filename">
- vcard.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<card>
  <name>Sachin Tendlya</name>
  <title>CEO, Master Blaster Inc.</title>
  <email>sachin.tendyla@masterblaster.com</email>
  <phone>(022) 333-4567</phone>
</card>
```

- Corresponding vcard.dtd

```
<!ELEMENT card (name,title,email,phone)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

DTD Components

- DTD is made up of following components
  - Elements
  - Attributes
  - Entities
  - PCDATA(parsed character data)
  - CDATA
- Predefined entities in XML

Entity References	Character
<	<
>	>
&	&
"	"
'	'

Declaring Elements

- Declaring Elements

```
<!ELEMENT card (name,title,email,phone)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

- Declaring minimum one occurrence

```
<!ELEMENT element-name (child-name+)>
```

- Declaring zero or more occurrences

```
<!ELEMENT element-name (child-name*)>
```

XML Schema

- XML Schema is an XML-based alternative to DTDs
- It describes the structure of an XML document
- The XML Schema language is also referred to as XML Schema Definition (XSD)
- Advantages
  - XML Schemas Support Data Types
  - XML Schemas use XML Syntax

Restrictions for DataTypes

Constraint	Description
enumeration	Defines a list of acceptable values
fractionDigits	Specifies the maximum number of decimal places allowed. Must be equal to or greater than zero
length	Specifies the exact number of characters or list items allowed. Must be equal to or greater than zero
maxExclusive	Specifies the upper bounds for numeric values (the value must be less than this value)
maxInclusive	Specifies the upper bounds for numeric values (the value must be less than or equal to this value)
maxLength	Specifies the maximum number of characters or list items allowed. Must be equal to or greater than zero
minExclusive	Specifies the lower bounds for numeric values (the value must be greater than this value)
minInclusive	Specifies the lower bounds for numeric values (the value must be greater than or equal to this value)
minLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero
pattern	Defines the exact sequence of characters that are acceptable
totalDigits	Specifies the exact number of digits allowed. Must be greater than zero
whiteSpace	Specifies how white space (line feeds, tabs, spaces, and carriage returns) is handled

Formatting XML Documents

- CSS stands for Cascading Style Sheets
- CSS is a language that you can use to define styles against any HTML element.
- These styles are set using CSS *properties*.

Implementing CSS

- Four ways to implement
  - Inline CSS
  - Embedded CSS
  - External CSS
  - Imported CSS

CSS Advantages

- CSS saves time
- Pages load faster
  - Less code means faster download times.
- Easy maintenance
  - To change the style of an element, you only have to make an edit in one place.
- Superior styles to HTML /XML
  - CSS has a much wider array of attributes than HTML.

XSL Languages

- XSL stands for E**X**tensible **S**tylesheet **L**anguage
- CSS = Style Sheets for HTML
  - HTML uses predefined tags, and the meaning of each tag is **well understood**
- XSL = Style Sheets for XML
  - XSL describes how the XML document should be displayed
- XSL consists of three parts:
  - XSLT - a language for transforming XML documents
  - XPath - a language for navigating in XML documents
  - XSL-FO - a language for formatting XML documents

XSLT

- Language for transforming XML documents into other formats .
  - XSLT stands for XSL Transformations
  - XSLT is the most important part of XSL
  - XSLT transforms an XML document into another XML document
  - XSLT uses XPath to navigate in XML documents
  - XSLT is a W3C Recommendation

XSLT Syntax

- Correct way to declare a XSL Stylesheet

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

OR

```
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

- An XSL style sheet consists of one or more set of rules that are called templates.
- The <xsl:template> Element
- The <xsl:value-of> Element

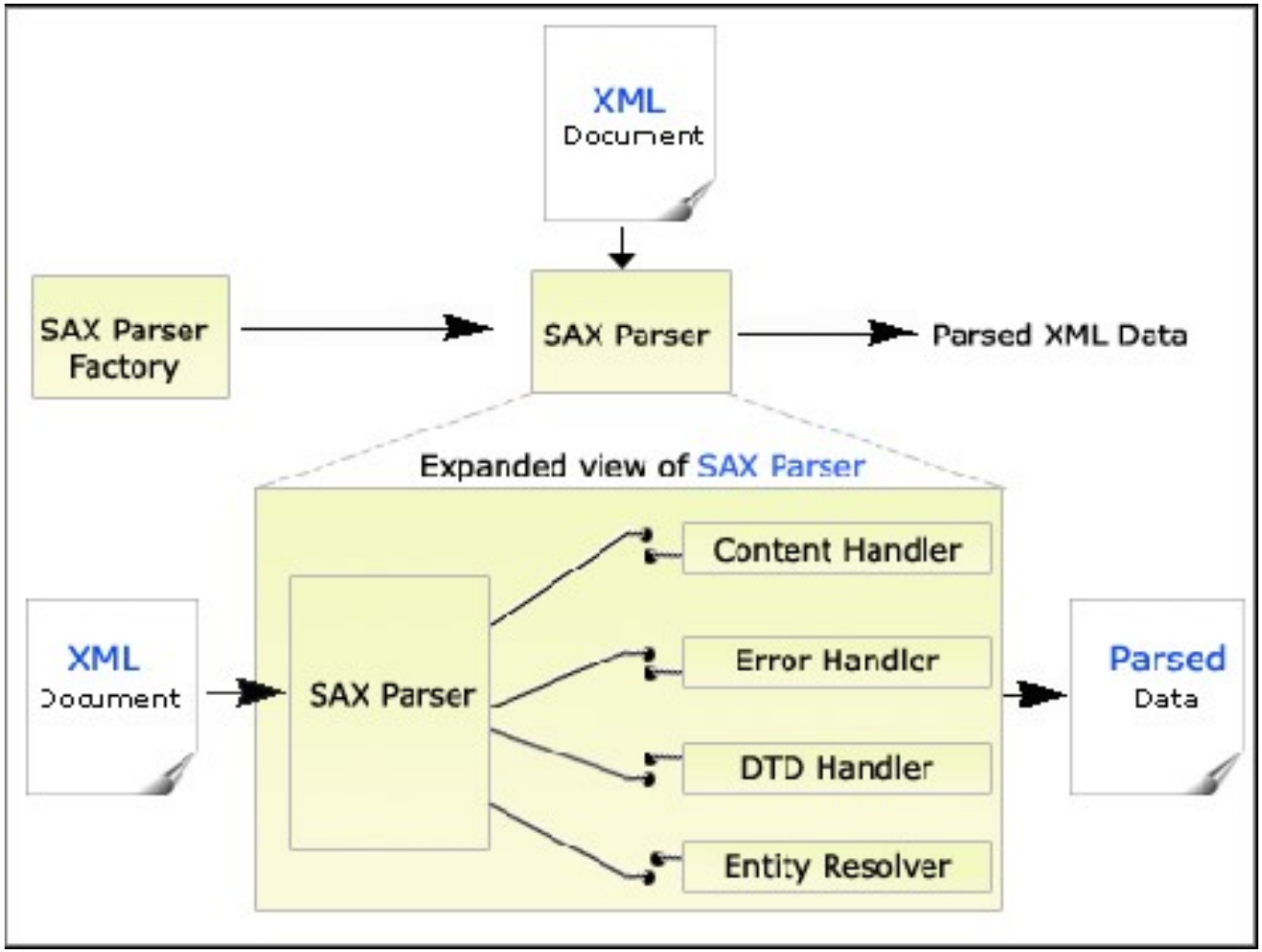
XML Parsers

XML parser is used to read, update, create and manipulate an XML document

- DOM – Document Object Model
  - Presents the document as a Tree Structure
  - Easy to manipulate and navigate to any element
- SAX – Simple API for XML
  - Event driven protocol
  - Less memory intensive
  - Generally used for large documents

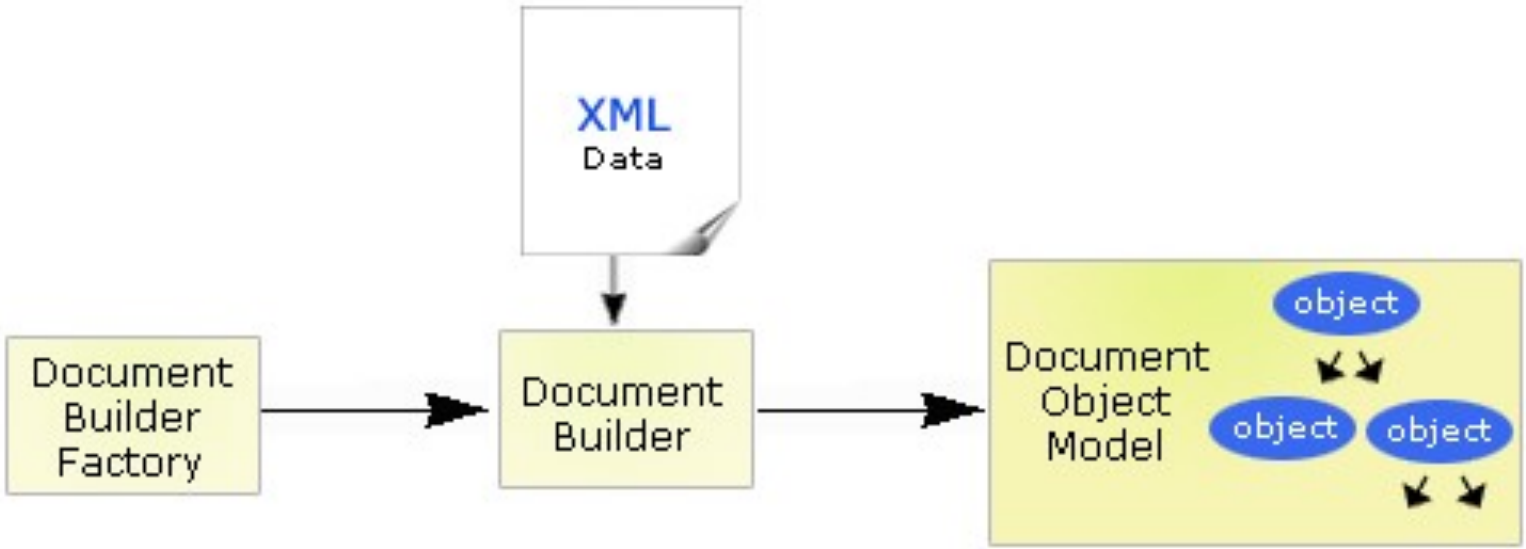
SAX Parser

- SAX architecture



DOM Parser

- DOM Architecture



XML Applications

- Used in Web Feeds like RSS, ATOM
- Used in SOAP
- Used in SAP Applications
- Used in Data Transfer
- Used in Framework Applications

XML and SOAP

- SOAP is a simple XML-based protocol to let applications exchange information over HTTP
  - SOAP stands for Simple Object Access Protocol
  - SOAP is a communication protocol
  - SOAP is a format for sending messages
  - SOAP communicates via Internet
  - SOAP is platform independent
  - SOAP is language independent
  - SOAP is based on XML
  - SOAP is simple and extensible
  - SOAP is a W3C recommendation

XML and RSS/ATOM

- RSS (originally RDF Site Summary, often dubbed *Really Simple Syndication*
  - An RSS document (which is called a "web feed") includes full or summarized text, plus metadata such as publishing dates and authorship
- Atom was started as an alternative to RSS
  - The *Atom Syndication Format* is an XML language used for web feeds
  - *Atom Publishing Protocol* (*AtomPub* or *APP*) is a simple HTTP-based protocol for creating and updating web resources.