

COP 4610L: Applications in the Enterprise Spring 2008

Introduction to PHP – Part 1

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
<http://www.cs.ucf.edu/courses/cop4610L/spr2008>

School of Electrical Engineering and Computer Science
University of Central Florida



Introduction to PHP

- PHP is officially known as PHP: Hypertext Preprocessor and is very rapidly becoming the most popular server-side scripting language for creating dynamic web pages.
- PHP was created in 1994 by Rasmus Lerdorf (who currently works for Linuxcare, Inc. as a senior open-source researcher) to track users at his Web site. Lerdorf originally called it Personal Home Page Tools in a package he released in 1995. It eventually became an Apache Software Foundation project.
- PHP2 featured built-in database support and form handling. In 1997, PHP3 was released and featured a new parser which substantially increased performance and led to an explosion in PHP use.



Introduction to PHP (cont.)

- PHP4 featured the Zend Engine and was considerably faster and more powerful than its predecessors and further enhanced the popularity of PHP.
- The current release is PHP 5.2.5 and features the Zend Engine 2, which provides further increases in speed and functionality. You can download the latest version of PHP at www.php.net. For more details on the Zend Engine 2 see www.zend.com.
- Today more than 17 million domains utilize PHP technology.
- All of the examples we'll be looking at use the latest stable version of PHP which is 5.2.5 and was released November 8, 2007.



Introduction to PHP (cont.)

- The power of the Web resides not only in serving content to users, but also in responding to requests from users and generating Web pages with dynamic content.
- Interactivity between the user and the server has become a crucial part of Web functionality. While other languages can also perform these functions, PHP was written specifically for interacting with the Web.
- PHP code is embedded directly into XHTML documents. This allows the document author to write XHTML in a clear, concise manner, without having to use multiple `print` statements, as is necessary with other CGI-based languages.



Introduction to PHP (cont.)

- PHP script file names usually end with `.php`, although a server can be configured to handle other file extensions.
- To run a PHP script, PHP must first be installed on your system. Download PHP 5.2.5 from www.php.net. (Most recent version is 5.2.5.)
- Although PHP can be used from the command line, a Web server is required to take full advantage of the scripting language. I would suggest the Apache server available from www.apache.org. (Note: this is not the Tomcat server you've already used.) Current version is 2.2.8 which is a new major version change from the previous 2.0.xx versions (mostly in the areas of security).
- The easiest way to get this setup is to use WAMP Server. The current version of this is WAMP 2.0 which automatically loads and configures Apache 2.2.8, MySQL 5.0.51 and PHP 5.2.5 This is how I'll show you to get it set-up. Go to www.wampserver.com.



Install PHP 5 Apache MySQL on Windows : WampServer - Windows Internet Explorer

http://www.wampserver.com/en/

File Edit View Favorites Tools Help

Google G Go Bookmarks

AutoFill Send to Settings

Page Tools

WampServer

Apache, PHP, MySQL on Windows

HOMEPAGE PRESENTATION **DOWNLOADS** ADDONS FORUM FAQ CREDITS DONATIONS ANASKA BLOG

Apache, MySQL, PHP on Windows

FR GB

LATEST RELEASE

WampServer 2.0 [11/21/2007]
Includes :
- Apache 2.2.6
- MySQL 5.0.45
- PHP 5.2.5

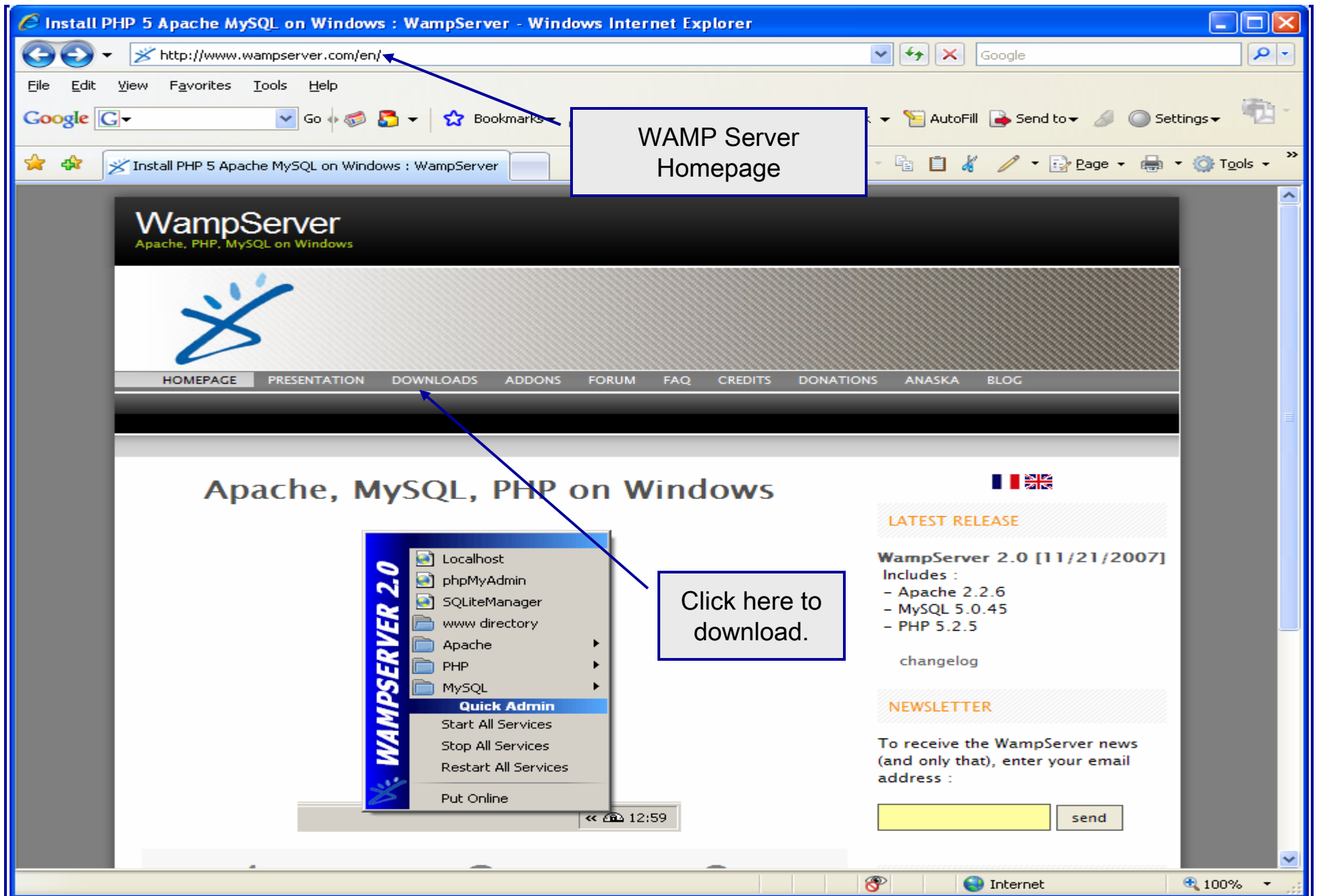
changelog

NEWSLETTER

To receive the WampServer news (and only that), enter your email address :

12:59

Internet 100%



Download WampServer 2 - Download PHP, apache, MySQL - Windows Internet Explorer

http://www.wampserver.com/en/download.php

File Edit View Favorites Tools Help


Google G Go Bookmarks 87 blocked Check AutoLink AutoFill Send to Settings

Download WampServer 2 - Download PHP, apache, M...

Home RSS Print Page Tools

WampServer

Apache, PHP, MySQL on Windows



[HOMEPAGE](#)
[PRESENTATION](#)
[DOWNLOADS](#)
[ADDONS](#)
[FORUM](#)
[FAQ](#)
[CREDITS](#)
[DONATIONS](#)
[ANASKA](#)
[BLOG](#)

Downloads

WampServer is an open source project, free to use (GPL licence). If you think our work deserves it and you want to help us, you can make a donation with paypal.

WARNING : do not try to install WampServer 2 over WAMP5. If WAMP5 is installed on your computer, save your data, uninstall it and delete the WAMP5 directory before installing WampServer 2.

[LATEST RELEASE](#)
WampServer 2.0 [11/21/2007]
 Includes :
 - Apache 2.2.6
 - MySQL 5.0.45
 - PHP 5.2.5
[changelog](#)

[NEWSLETTER](#)
 To receive the WampServer news (and only that), enter your email address :

DOWNLOAD WampServer 2.0
(November 21 2007)
Apache 2.2.6
PHP 5.2.5 + PECL
SQLitemanager
MySQL 5.0.45

http://www.wampserver.com/dl.php

Internet 100%

Download page

DOWNLOAD WampServer 2.0
(November 21 2007)
Apache 2.2.6
PHP 5.2.5 + PECL
SQLitemanager
MySQL 5.0.45

WAMP5 Homepage - Windows Internet Explorer


http://localhost/

File Edit View Favorites Tools Help

Google G Go Bookmarks 20 blocked Check AutoLink AutoFill Send to Settings

WAMP5 Homepage

Intranet settings are now turned off by default. Intranet settings are less secure than Internet settings. Click for options...



Version 2.0 [Version Française](#)

Server Configuration

Apache Version : 2.2.8

PHP Version : 5.2.5

Loaded Extensions :

✳ bcmath	✳ calendar	✳ com_dotnet	✳ ctype	✳ session
✳ filter	✳ ftp	✳ hash	✳ iconv	✳ json
✳ odbc	✳ pcre	✳ Reflection	✳ date	✳ libxml
✳ standard	✳ tokenizer	✳ zlib	✳ SimpleXML	✳ dom
✳ SPL	✳ wddx	✳ xml	✳ xmlreader	✳ xmlwriter
✳ apache2handler	✳ gd	✳ mbstring	✳ mysql	✳ mysqli
✳ PDO	✳ pdo_mysql	✳ SQLite		

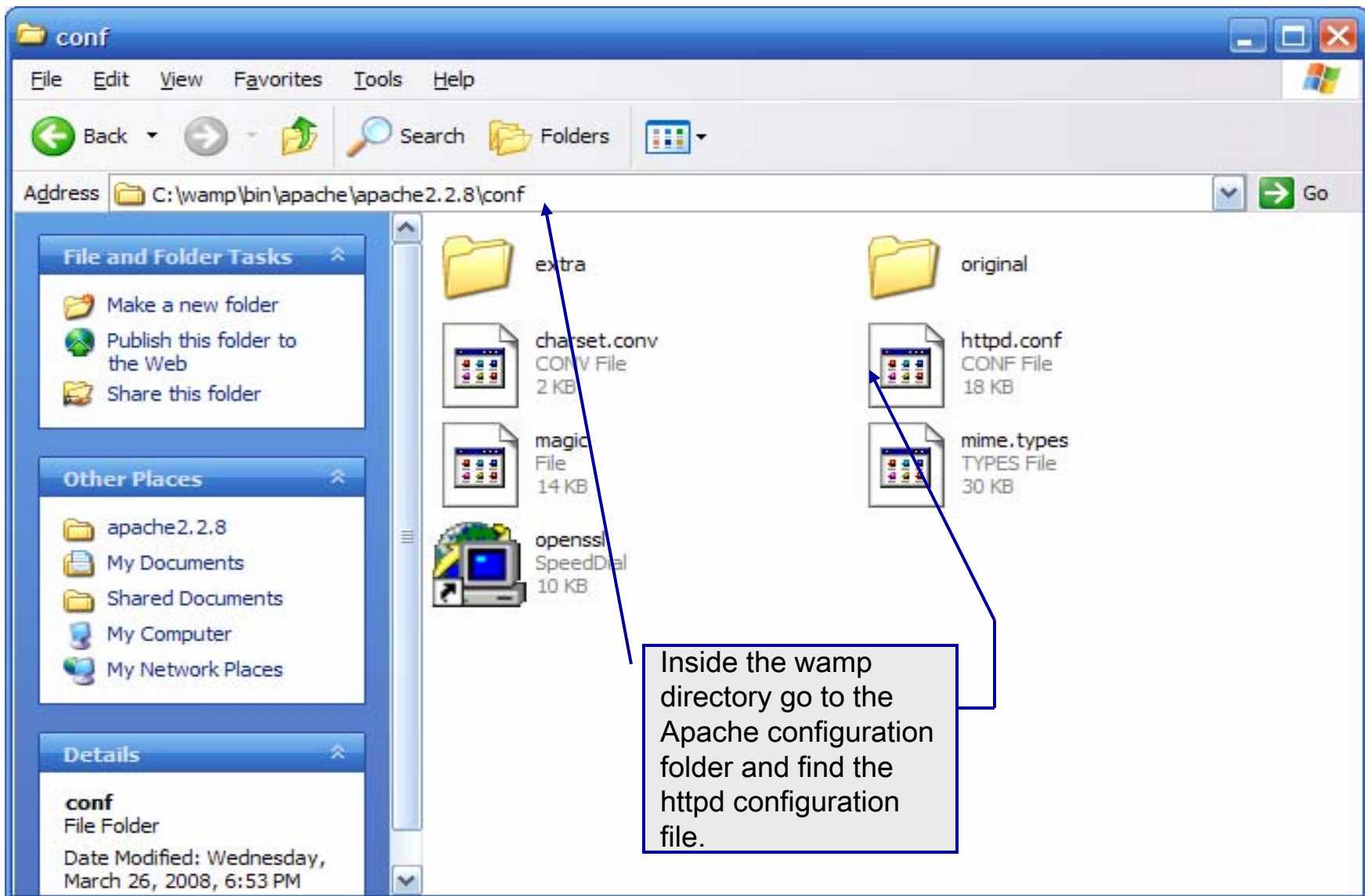
MySQL Version : 5.0.51a

Tools

- [phpinfo\(\)](#)
- [phpmyadmin](#)

Done Internet 100%





```
File Edit Format View Help
# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum number of requests a server process serves
ThreadsPerChild 250
MaxRequestsPerChild 0

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to point the LockFile directive
# at a local disk. If you wish to share the same ServerRoot for multiple
# httpd daemons, you will need to change at least LockFile and PidFile.
#
ServerRoot "E:/Program Files/wamp/apache2"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0
#
#Listen 12.34.56.78:80
Listen 8081

#
# Dynamic shared object (DSO) support
#
# To be able to use the functionality of a module which was built as a DSO you
```

Edit the httpd configuration file to have the Apache server listen on port 8081 instead of port 80 as is the default case.



A PHP Test Example

Create this file named `hello.php` and save it to the `htdocs` folder in Apache. Then start the Apache server, enter the URL: <http://localhost:8081/hello.php> and you should see output similar to that shown on the next slide.

```
<html>
<head>
<title>Hello From PHP</title>
</head>
<body style = "font-family: arial, sans-serif;
background-color: #856363" background=image1.jpg>
<h1> Hello From PHP</h1>
```

```
<?php
print "Current Information";
phpInfo();
?>
```

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


This is
PHP



Windows Internet Explorer window titled "Hello From PHP - Windows Internet Explorer". The address bar shows "http://localhost/hello.php". The page content includes:

Hello From PHP

Current Information

PHP Version 5.2.5 

System	Windows NT MARK-PC2 5.1 build 2600
Build Date	Nov 8 2007 23:18:08
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" --with-gd
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\wamp\bin\apache\apache2.2.8\bin\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled

A blue callout box with a white background and a blue border contains the text: "The default directory for the php.ini file. Set by WAMP." A blue arrow points from this box to the "Loaded Configuration File" row in the table above.



A First PHP Example

- The following two pages illustrate a simple PHP “hello world” program.
- In PHP, code is inserted between the scripting delimiters `<?php` and `?>`. PHP code can be placed anywhere in XHTML markup, as long as the code is enclosed in these scripting delimiters.
- Place all of your XHTML and PHP files inside the WAMP www directory.



welcome.php Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- welcome.php -->
<!-- XHTML file containing a PHP script. -->

<?php
    $name = "Mark";    //php declaration and assignment
?>

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

  <!-- head section of document -->
  <head>
    <title>A Simple PHP Document</title>
  </head>

  <!-- body section of document -->
  <body style = "font-size: 2em">
    <hr>
    <font color = blue><h1> Generating HTML From PHP </h1></font color>
    <p>
```

PHP code
declaring a
variable.



welcome.php Example

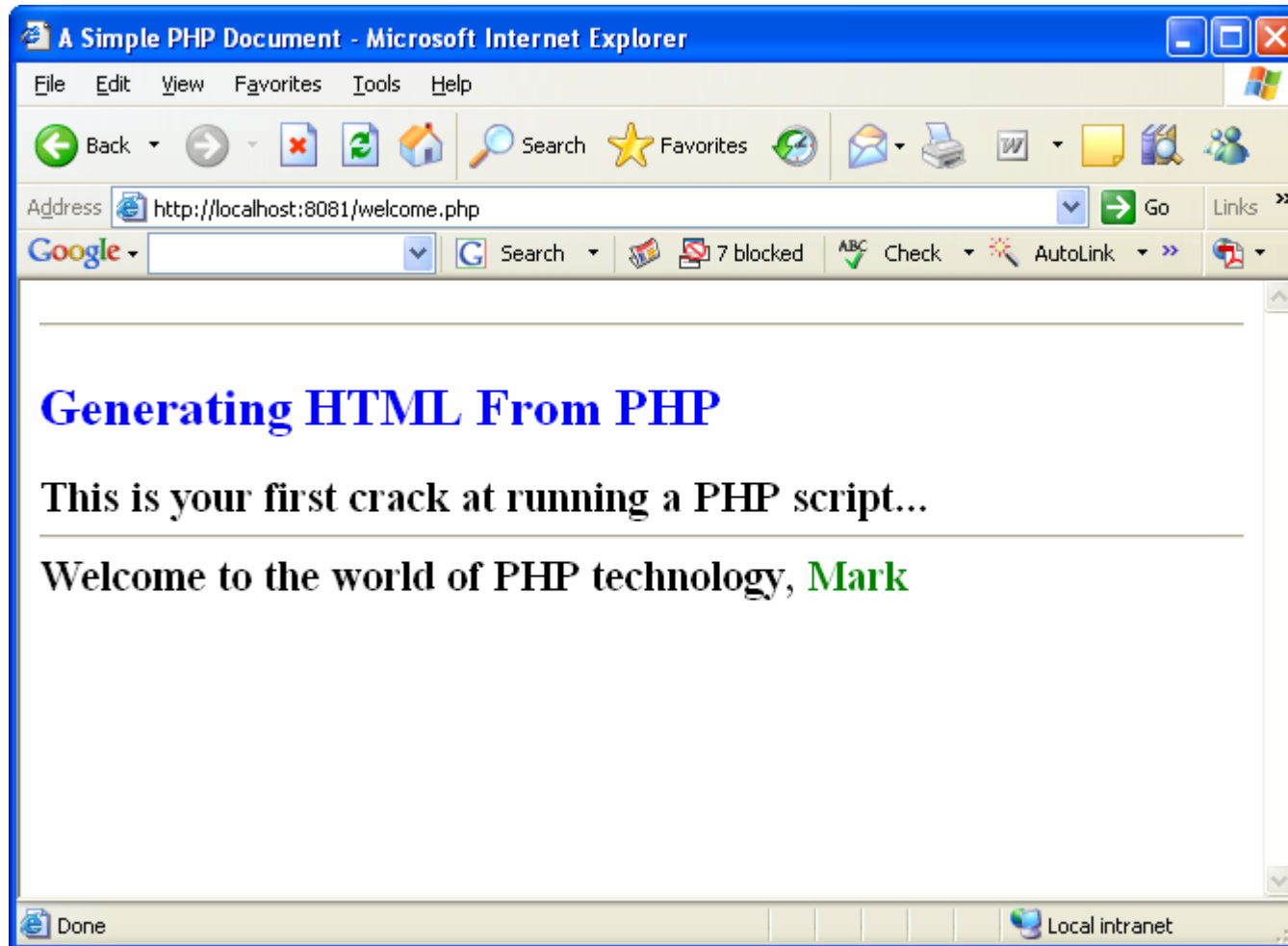
```
<strong>
  <!--print variable name's value in the message-->
  {
  <?php
    print("This is your first crack at running a PHP script...");
    print("<HR>");
    print("Welcome to the world of PHP technology, ");
  ?>
  <font color = green>
  {
  <?php
    print("$name");
  ?>
  </font color>
  }
  </strong>
</p>
</body>
</html> <!-- end XHTML document -->
```

PHP code

PHP code



welcome.php Example Output



Viewing Client/Server Environment Variables

- Knowledge of a client's execution environment is useful to system administrators who want to provide client-specific information.
- Environment variables contain information about a script's environment, such as the client's web browser, the HTTP host and the HTTP connection.
 - The table on the next page summarizes some of the superglobal arrays defined by PHP.
- The XHTML document on page 19 displays the values of the server's environment variables in a table. PHP stores the server variables and their values in the `$_SERVER` array. Iterating through the array allows one to view all of the server's environment variables.



Some Superglobal Environment Arrays

Variable Name	Description
<code>\$_SERVER</code>	Data about the currently running server.
<code>\$_ENV</code>	Data about the client's environment.
<code>\$_GET</code>	Data posted to the server by the <code>get</code> method.
<code>\$_POST</code>	Data posted to the server by the <code>post</code> method.
<code>\$_COOKIE</code>	Data contained in cookies on the client's computer.
<code>\$GLOBALS</code>	Array containing all global variables.



server.php Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- server.php -->
<!-- Program to display $_SERVER variables -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title>SERVER Variables Display</title>
  </head>

  <body style = "font-family: arial, sans-serif;
  background-color: #856363" background=image1.jpg>

  <table border = "0" cellpadding = "2" cellspacing = "0"
  width = "100%">
  <?php
  // print the key and value for each element
  // in the $_SERVER array
  foreach ( $_SERVER as $key => $value )
    print( "<tr><td bgcolor = \"#11bbff\">
    <strong>$key</strong></td> <td>$value</td></tr>" );
  ?>
  </table>
</body>
</html>
```

Iterate through the
\$_SERVER array to list all
of the SERVER variables for
the current server on which
PHP is running.



SERVER Variables Display - Windows Internet Explorer

http://localhost/server.php

Output from executing server.php

Norton 360 Fraud monitoring is on

HTTP_ACCEPT	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
HTTP_ACCEPT_LANGUAGE	en-us
HTTP_UA_CPU	x86
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.0.3705; .NET CLR 1.1.4322; Media Center PC 4.0)
HTTP_HOST	localhost
HTTP_CONNECTION	Keep-Alive
PATH	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;c:\Python22;C:\Program Files\QuickTime\QTSystem\
SystemRoot	C:\WINDOWS
COMSPEC	C:\WINDOWS\system32\cmd.exe
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
WINDIR	C:\WINDOWS
SERVER_SIGNATURE	
SERVER_SOFTWARE	Apache/2.2.8 (Win32) PHP/5.2.5
SERVER_NAME	localhost
SERVER_ADDR	127.0.0.1
SERVER_PORT	80
REMOTE_ADDR	127.0.0.1
DOCUMENT_ROOT	C:/wamp/www/

Done Internet 100%



Form Processing and Business Logic

- XHTML forms enable web pages to collect data from users and send it to a web server for processing.
- Interaction of this kind between users and web servers is vital to e-commerce applications. Such capabilities allow users to purchase products, request information, send and receive web-based email, perform on-line paging and take advantage of various other online services.
- The XHTML document on the next few pages collects information from a user for the purposes of adding them to a mailing list.
- The PHP file on page 23 validates the data entered by the user through the form and “registers” them in the mailing list database.



form.html Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!-- form.html -->
```

```
<!-- Form for use with the form.php program -->
```

This XHTML document generates the form that the user will submit to the server via form.php

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

```
<head>
```

```
<title>Sample form to take user input in XHTML</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a sample registration form.</h1>
```

Please fill in all fields and click Register.

```
<!-- post form data to form.php -->
```

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

```
<img src = "images/user.gif" alt = "User" /><br />
```

```
<span style = "color: blue">
```

```
  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 = "Products" /><br />

<span style = "color: blue">
  Which publication would you like information about?
</span><br />

<!-- create drop-down list containing magazine names -->
<select name = "magazine">
  <option>Velo-News</option>
  <option>Cycling Weekly</option>
  <option>Pro Cycling</option>
  <option>Cycle Sport</option>
  <option>RadSport</option>
  <option>Mirror du Cyclisme</option>
</select>
<br /><br />
```



```
<img src = "images/os.gif" alt = "Operating System" />
<br /><span style = "color: blue">
  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 2000" />
  Windows 2000
<input type = "radio" name = "os" value =
  "Windows 98" />
  Windows 98<br />
<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" />
</form>

</body>
</html>
```



form.php Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!-- form.php -->
```

```
<!-- Read information sent from form.html -->
```

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

```
<head>
```

```
<title>Form Validation</title>
```

```
</head>
```

```
<body style = "font-family: arial,sans-serif">
```

```
<?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 style = \"color: red; font-size: 2em\">
```

```
INVALID PHONE NUMBER:</span><br />
```

```
A valid phone number must be in the form
```

```
<strong>(555)555-5555</strong><br />
```

```
<span style = \"color: blue\">
```

```
Click the Back button, enter a valid phone number and resubmit.<br /><br />
```

```
Thank You.</span></p></body></html> " );
```

```
die(); // terminate script execution
```

```
}
```

```
?>
```

Function extract (associativeArray) creates a variable-value pair corresponding to each key-value pair in the associative array \$_POST.

See page 36 for explanation of regular expressions.

Function die() terminates script execution. An error has occurred, no need to continue.



```

<p>Hi
  <span style = "color: blue"> <strong> <?php print( "$fname" ); ?> </strong> </span>.
  Thank you for completing the survey.<br />
  You have been added to the <span style = "color: blue">
    <strong> <?php print( "$magazine " ); ?> </strong> </span> mailing list.
</p>
<strong>The following information has been saved in our database:</strong><br />
<table border = "0" cellpadding = "0" cellspacing = "10">
  <tr>
    <td bgcolor = "#ffffaa">Name </td>
    <td bgcolor = "#ffffbb">Email</td>
    <td bgcolor = "#ffffcc">Phone</td>
    <td bgcolor = "#ffffdd">OS</td>
  </tr>
  <tr>
    <?php
      // print each field's value
      print( "<td>$fname $lname</td> <td>$email</td> <td>$phone</td> <td>$os</td>" );
    ?>
  </tr>
</table>
<br /><br /><br />
<div style = "font-size: 10pt; text-align: center">
  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 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8081/form.html> Go Links >>

This is a sample registration form.

Please fill in all fields and click Register.

User Information
Please fill out the fields below.

First Name

Last Name

Email

Phone

Must be in the form (555)555-5555

Publications
Which publication would you like information about?

Velo-News

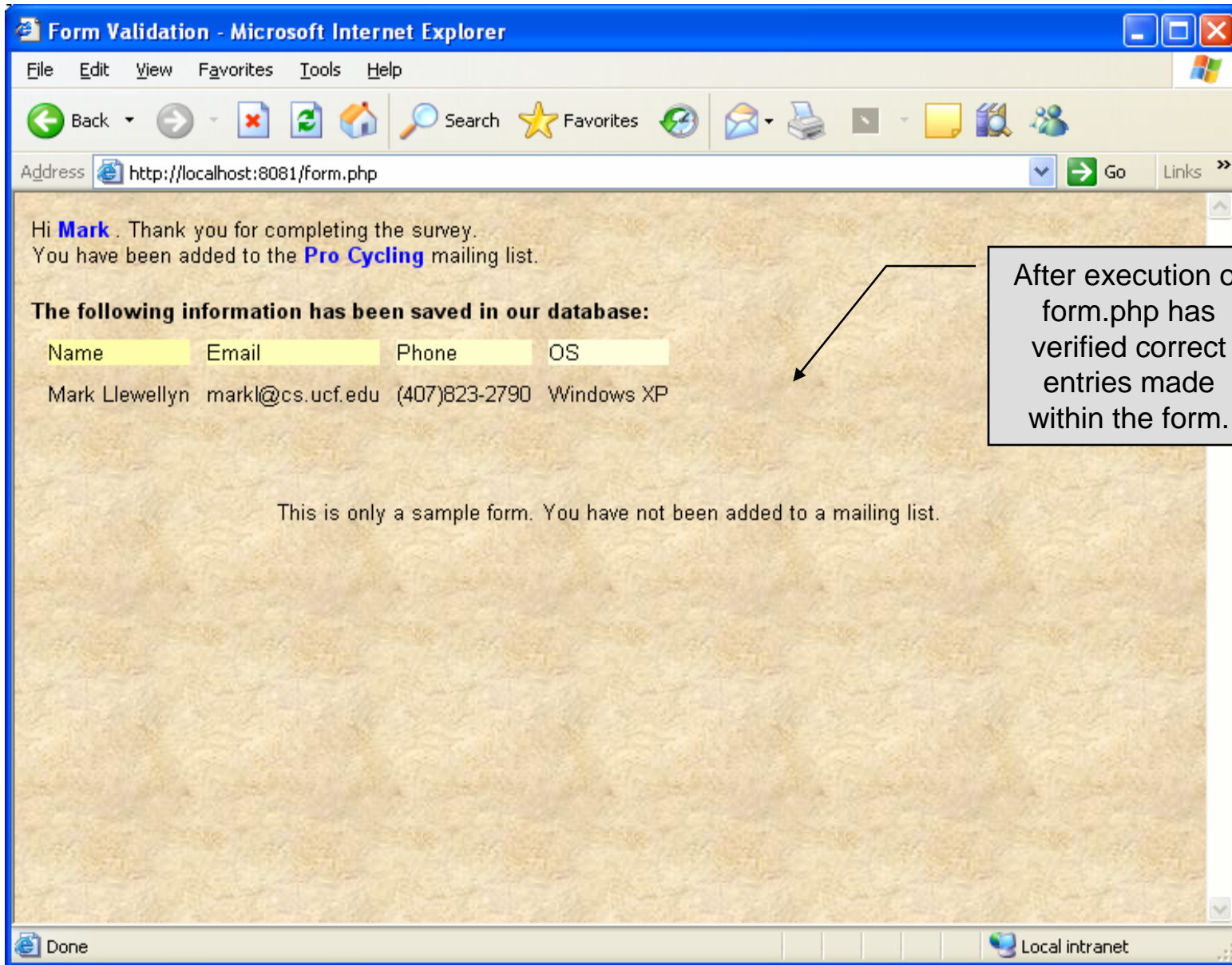
Operating System
Which operating system are you currently using?

Windows XP Windows 2000 Windows 98
 Linux Other

Done Local intranet

Execution of form.html within a web browser





Sample form to take user input in XHTML - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <http://localhost:8081/form.html> Go Links

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name

Last Name

Email

Phone

Must be in the form (555)555-5555

Publications

Which publication would you like information about?

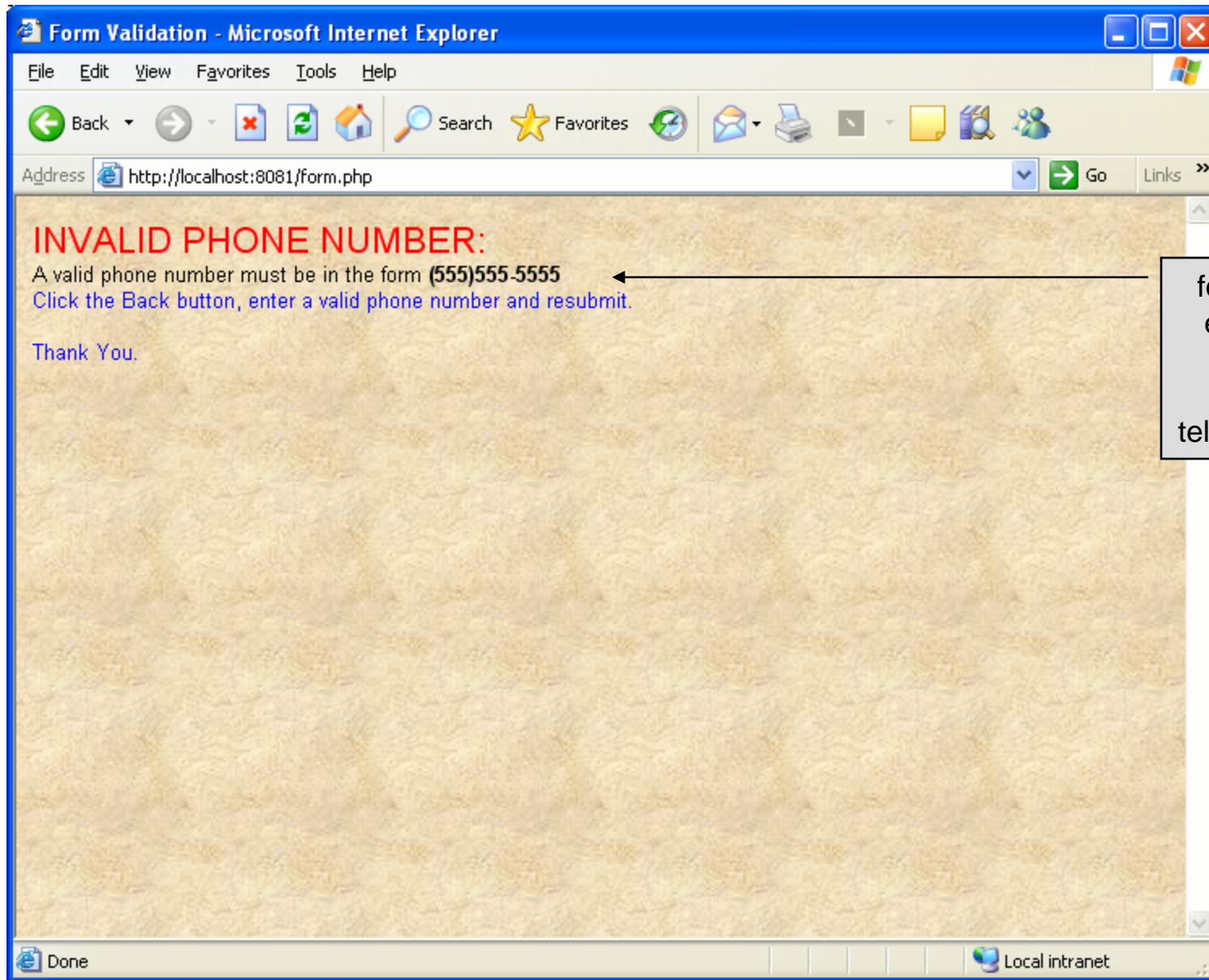
Operating System

Which operating system are you currently using?

Windows XP Windows 2000 Windows 98
 Linux Other

User enters an improperly formatted telephone number in the form.





form.php issues error regarding improperly formatted telephone number.



How the Form Example Works

- The `action` attribute of the form element, indicates that when the user clicks the `Register` button, the form data will be posted to `form.php` for processing.
- Using `method = "post"` appends the form data to the browser request that contains the protocol (i.e., HTTP) and the requested resource's URL. Scripts located on the web server's machine (or accessible through the network) can access the form data sent as part of the request.
- Each of the form's input fields are assigned a unique name. When `Register` is clicked, each field's name and value are sent to the web server.
- Script `form.php` then accesses the value for each specific field through the global array `$_POST`.



How the Form Example Works (cont.)

- The superglobal arrays are associative arrays predefined by PHP that hold variable acquired from the user input, the environment, or the web server and are accessible in any variable scope.
 - If the information from the form had been submitted via the HTTP method `get`, then the superglobal array `$_GET` would contain the name-value pairs.
- Since the HTML form and the PHP script “communicate” via the name-value pairs, it is a good idea to make the XHTML object names meaningful so that the PHP script that retrieves the data is easier to understand.



Register_globals

- In PHP versions 4.2 and higher, the directive `register_globals` is set to `Off` by default for security reasons.
- Turning off `register_globals` means that all variables sent from an XHTML form to a PHP document now must be accessed using the appropriate superglobal array (either `$_POST` or `$_GET`).
- When this directive was turned On, as was the default case in PHP versions prior to 4.2, PHP created an individual global variable corresponding to each form field.



Validation of Form Generated Data

- The form example illustrates an important concept in the validation of user input. In this case, we simply checked the validity of the format of the telephone number entered by the client user.
- In general, it is crucial to validate information that will be entered into database or used in mailing lists. For example, validation can be used to ensure that credit-card numbers contain the proper number of digits before the numbers are encrypted to a merchant.
- In this case, the form.php script is implementing the **business logic** or **business rules** for our application.



Pattern Matching in PHP

- For powerful string comparisons (pattern matching), PHP provides functions `ereg` and `preg_match`, which use regular expressions to search a string for a specified pattern.
- Function `ereg` uses **Portable Operating System Interface (POSIX) extended regular expressions**.
 - POSIX-extended regular expressions are a standard to which PHP regular expression conform.
- Function `preg_match` provides **Perl-compatible regular expressions**.
- Perl-compatible regular expressions are more widely used than POSIX regular expressions. PHP's support for Perl-compatible regular expressions eases migration from Perl to PHP. The following examples illustrate these concepts.



expression.php - Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- expression.php -->
<!-- Using 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 function 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 />" );
```

^ matches at beginning
of a string

\$ matches at end of a
string



```

// 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 );
}

print( "<br />" );
?>
</body>
</html>

```

Uses a regular expression to match a word ending in "ow".



Output From `expression.php` - Example



Verifying a Username and Password Using PHP

- It is often the case that a private website is created which is accessible only to certain individuals.
- Implementing privacy generally involves username and password verification.
- In the next example, we'll see an XHTML form that queries a user for a username and password. The fields `USERNAME` and `PASSWORD` are posted to the PHP script `verify.php` for verification.
 - For simplicity, data is not encrypted before sending it to the server.
 - For more information on PHP encryption functions visit: <http://www.php.net/manual/en/ref.mcrypt.php>.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- password.html -->
<!-- XHTML form sent to password.php for verification -->

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title>Verifying a username and a password.</title>
    <style type = "text/css">
      td { background-color: #DDDDDD }
    </style>
  </head>
  <body style = "font-family: arial">
    <p style = "font-size: 18pt">
      <font color=red><B> Welcome to the COP 4610 High Security WebPage </B></font><HR>
    <p style = "font-size: 13pt">
      Type in your username and password below.
      <br />
      <span style = "color: #0000FF; font-size: 10pt;
        font-weight: bold">
        Note that password will be sent as plain text - encryption not used in this application
      </span>
    </p>
```




```
<!-- post form data to password.php -->
<form action = "password.php" method = "post">
  <br />
  <table border = "3" cellspacing = "3" style = "height: 90px; width: 150px;
  font-size: 10pt" cellpadding = "1">
    <tr>
      <td colspan = "3"> <strong>Username:</strong> </td>
    </tr>
    <tr>
      <td colspan = "3"> <input size = "40" name = "USERNAME"
      style = "height: 22px; width: 115px" /> </td>
    </tr>
    <tr>
      <td colspan = "3"> <strong>Password:</strong> </td>
    </tr>
    <tr>
      <td colspan = "3"> <input size = "40" name = "PASSWORD"
      style = "height: 22px; width: 115px" type = "password" /> <br/></td>
    </tr>
    <tr>
      <td colspan = "1">
        <input type = "submit" name = "Enter" value = "Enter" style = "height: 23px;
        width: 47px" /> </td>
      <td colspan = "2"> <input type = "submit" name = "NewUser" value = "New User"
      style = "height: 23px" />
      </td>
    </tr>
  </table> </form> <HR> </body> </html>
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- password.php -->
<!-- Searching a database for usernames and passwords. -->

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <?php
      extract( $_POST );
      // check if user has left USERNAME or PASSWORD field blank
      if ( !$USERNAME || !$PASSWORD ) {
        fieldsBlank();
        die();
      }
      // check if the New User button was clicked
      if ( isset( $NewUser ) ) {
        // open password.txt for writing using append mode
        if ( !( $file = fopen( "password.txt", "a" ) ) ) {

          // print error message and terminate script
          // execution if file cannot be opened
          print( "<title>Error</title></head><body>
            Could not open password file
            </body></html>" );
          die();
        }
      }
    }
  }
}
```



```
// write username and password to file and call function userAdded
fputs( $file, "$USERNAME,$PASSWORD\n" );
userAdded( $USERNAME );
}
else {

// if a new user is not being added, open file
// for reading
if ( !( $file = fopen( "password.txt", "r" ) ) ) {
    print( "<title>Error</title></head>
        <body>Could not open password file
        </body></html>" );
    die();
}

$userVerified = 0;

// read each line in file and check username and password
while ( !feof( $file ) && !$userVerified ) {

    // read line from file
    $line = fgets( $file, 255 );

    // remove newline character from end of line
    $line = chop( $line );

    // split username and password using comma delimited string
    $field = split( ",", $line, 2 );
```



```
// verify username
if ( $USERNAME == $field[ 0 ] ) {
    $userVerified = 1;

    // call function checkPassword to verify user's password
    if ( checkPassword( $PASSWORD, $field ) == true )
        accessGranted( $USERNAME );
    else
        wrongPassword();
}
}

// close text file
fclose( $file );

// call function accessDenied if username has not been verified
if ( !$userVerified )
    accessDenied();
}

// verify user password and return a boolean
function checkPassword( $userpassword, $filedata )
{
    if ( $userpassword == $filedata[ 1 ] )
        return true;
    else
        return false;
}
```



```
// print a message indicating the user has been added
function userAdded( $name ) {
    print( "<title>Thank You</title></head>
        <body style = \"font-family: arial;
        font-size: 1em; color: blue\">
        <strong>You have been added
        to the user list, $name. Please remember your password.
        <br />Enjoy the site.</strong>" );
}

// print a message indicating permission has been granted
function accessGranted( $name ) {
    print( "<title>Thank You</title></head>
        <body style = \"font-family: arial;
        font-size: 1em; color: blue\">
        <strong>Permission has been
        granted, $name. <br />
        Enjoy the site.</strong>" );
}

// print a message indicating password is invalid
function wrongPassword() {
    print( "<title>Access Denied</title></head>
        <body style = \"font-family: arial;
        font-size: 1em; color: red\">
        <strong>You entered an invalid
        password.<br />Access has
        been denied.</strong>" );
}
```



```
// print a message indicating access has been denied
function accessDenied() {
    print( "<title>Access Denied</title></head>
    <body style = \"font-family: arial;
    font-size: 1em; color: red\">
    <strong>
    You were denied access to this server.
    <br /></strong>" );
}

// print a message indicating that fields
// have been left blank
function fieldsBlank() {
    print( "<title>Access Denied</title></head>
    <body style = \"font-family: arial;
    font-size: 1em; color: red\">
    <strong>
    Please fill in all form fields.
    <br /></strong>" );
}
?>
</body>
</html>
```



The image shows two overlapping browser windows. The top window, titled "Verifying a username and a password. - Windows Internet Explorer", displays a page with the URL "http://localhost:8081/password.html". The page content includes a red heading "Welcome to the COP 4610 High Security WebPage", a blue instruction "Type in your username and password below", and a note "Note that password will be sent as plain text - end". Below this is a form with "Username:" (containing "Mark Llewellyn"), "Password:" (with masked characters), and buttons for "Enter" and "New User". A blue arrow points from the "New User" button to a text box on the right.

The bottom window, titled "Thank You - Windows Internet Explorer", displays a page with the URL "http://localhost:8081/password.php". The page content is a blue message: "You have been added to the user list, Mark Llewellyn. Please remember your password. Enjoy the site." A blue arrow points from the "New User" button in the top window to this second window.

Execution of password.html. Client-side XHTML form. User clicks on New User button to enter their information.

Execution of password.php to enter a new user.



The image displays two overlapping browser windows. The top window, titled "Verifying a username and a password. - Windows Internet Explorer", shows a page with the URL "http://localhost:8081/password.html". The page content includes a red heading "Welcome to the COP 4610 High Security WebPage", a text prompt "Type in your username and password below.", a blue note "Note that password will be sent as plain text - encryption not used in this application", and a form with "Username:" (containing "Mark Llewellyn"), "Password:" (with masked characters), and "Enter" and "New User" buttons. A blue arrow points from a text box to the "Enter" button. The bottom window, titled "Access Denied - Windows Internet Explorer", shows the URL "http://localhost:8081/password.php" and a red message "You entered an invalid password. Access has been denied." A blue arrow points from a text box to the URL bar.

Verifying a username and a password. - Windows Internet Explorer

http://localhost:8081/password.html

Google

Go

Bookmarks

14 blocked

Check

AutoLink

Verifying a username and a password.

Welcome to the COP 4610 High Security WebPage

Type in your username and password below.
Note that password will be sent as plain text - encryption not used in this application

Username:
Mark Llewellyn

Password:
.....

Enter New User

password.php

Access Denied - Windows Internet Explorer

http://localhost:8081/password.php

Google

Go

Bookmarks

Access Denied

**You entered an invalid password.
Access has been denied.**

Done Local intranet 100%

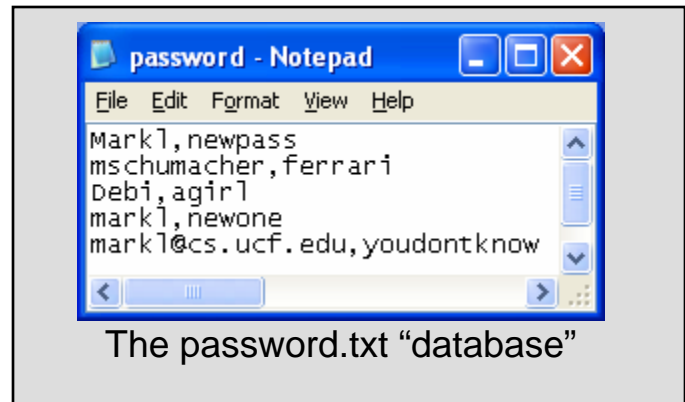
Execution of password.html. Client-side XHTML form. User clicks on Enter button to submit and verify their information.

Execution of password.php to invalidate an attempted entry by a user.



How password.php Works

- The PHP script `password.php` verifies the client's username and password by querying a database. For this example, the “database” of usernames and passwords is just a text file (for simplicity). Existing users are validated against this file, and new users are appended to it.
- Whether we are dealing with a new user is determined by calling function `isset` to test if variable `$NewUser` has been set.
- When the user submits the `password.html` form to the server, they click either **Enter** or **New User** button. After calling function `extract`, either variable `$NewUser` or `$Enter` is created depending on which button was selected. If `$NewUser` has not been set, we assume the user clicked **Enter**.

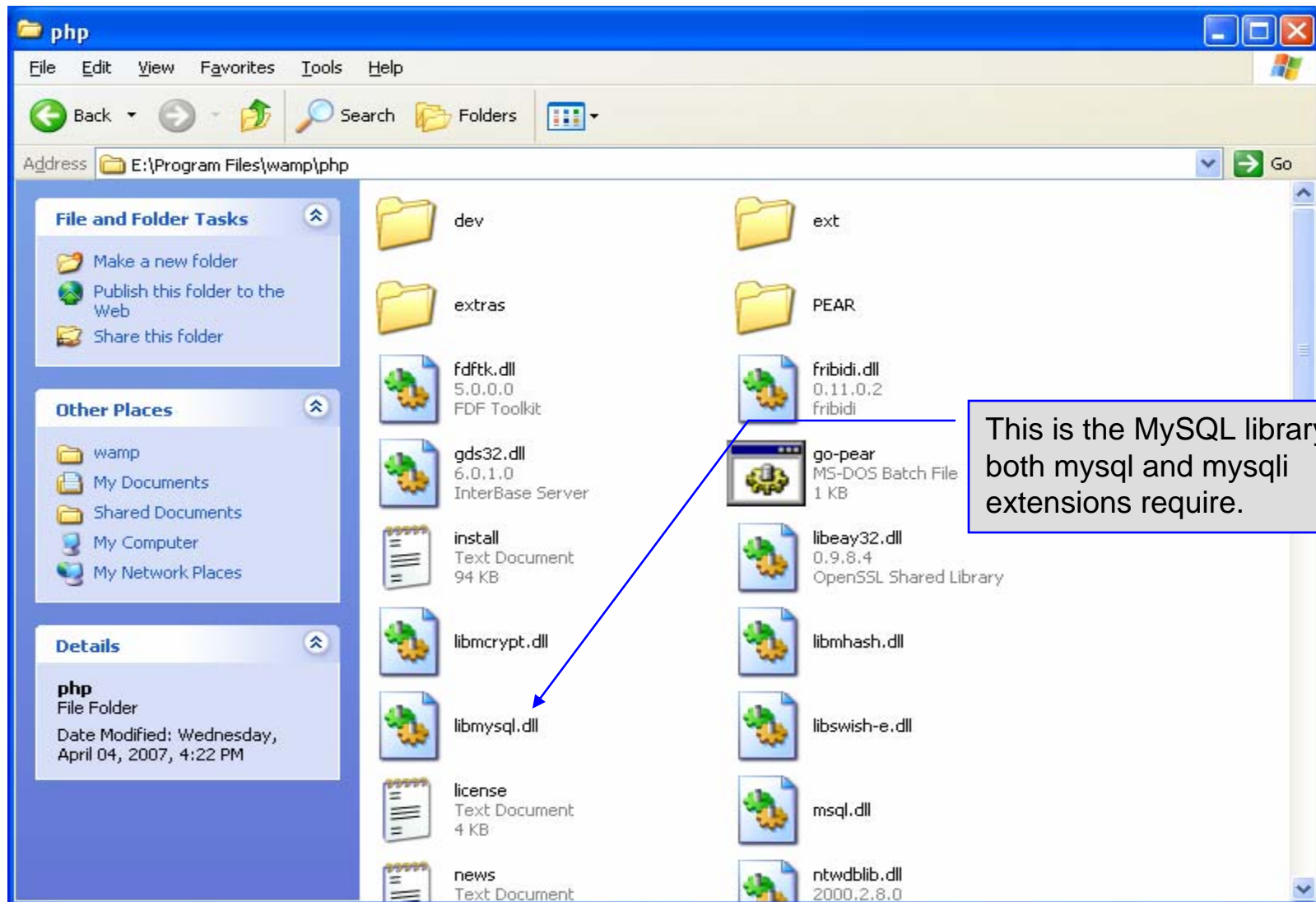


PHP and Database Connectivity

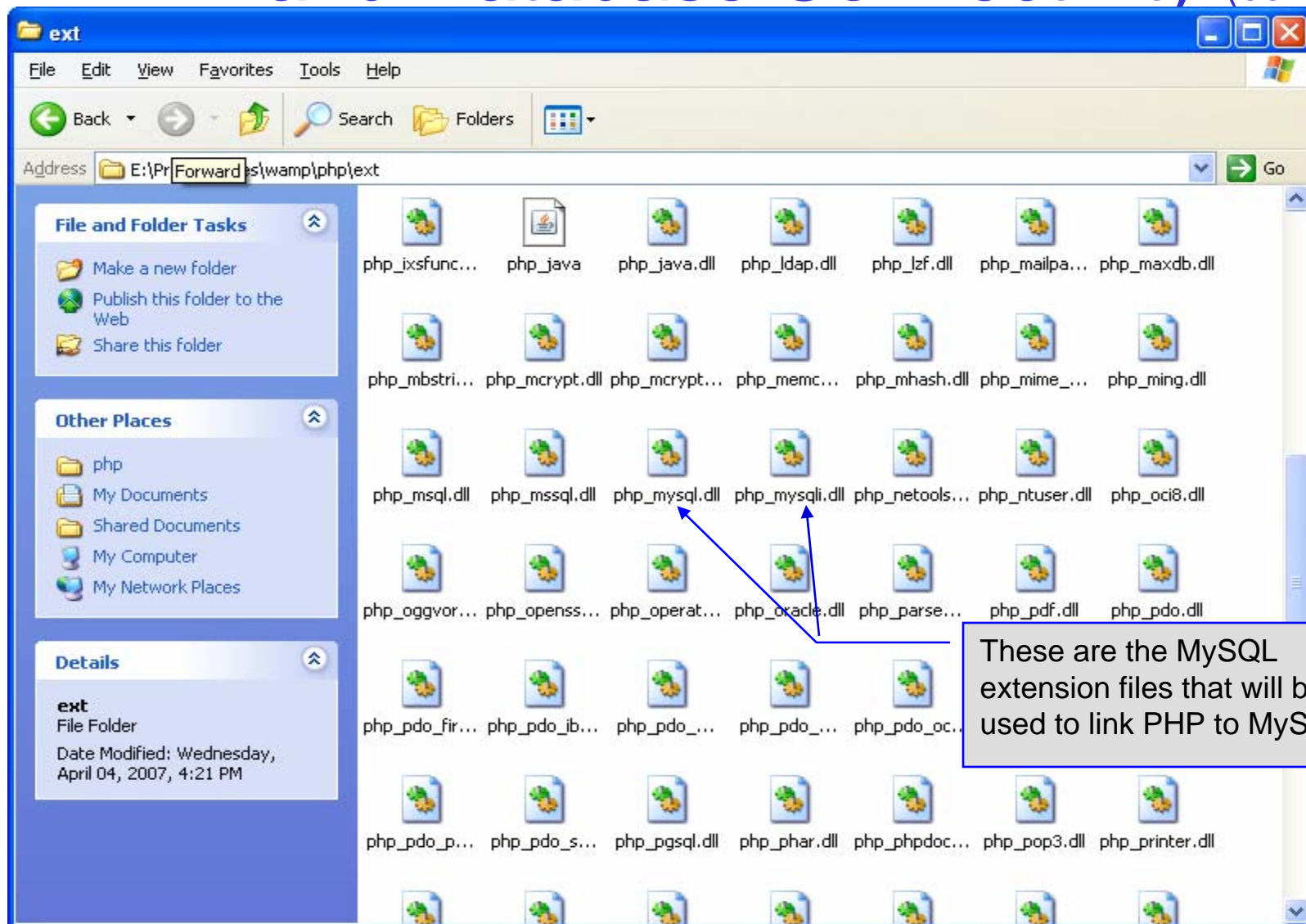
- PHP offers built-in support for a wide variety of database systems from Unix DBM through relational systems such as MySQL to full size commercial systems like Oracle.
- We'll continue to use MySQL as the underlying database system so that you can easily compare the work we've done with MySQL using Java servlets and JSPs.
- Before you go any further in these notes you must configure PHP to access MySQL databases. Beginning with PHP 5, MySQL is not enabled by default in PHP, nor is the MySQL library bundled with PHP.
 - Versions of MySQL greater than 4.1.0 use MySQLi extensions.
 - Versions of MySQL less than 4.1.0 use MySQL extensions.



PHP and Database Connectivity (cont.)



PHP and Database Connectivity (cont.)



PHP and Database Connectivity (cont.)

```
php - Notepad
File Edit Format View Help
;extension=php_ldap.dll
;extension=php_mcrypt.dll
;extension=php_mhash.dll
;extension=php_mime_magic.dll
;extension=php_ming.dll
;extension=php_mssql.dll
;extension=php_mysql.dll
extension=php_mysql.dll
extension=php_mysql.dll
;extension=php_oci8.dll
;extension=php_openssl.dll
;extension=php_oracle.dll
;extension=php_pdf.dll
;extension=php_pgsql.dll
;extension=php_sfmop.dll
;extension=php_snmp.dll
;extension=php_sockets.dll
;extension=php_sybase_ct.dll
;extension=php_tidy.dll
;extension=php_xmlrpc.dll
;extension=php_xsl.dll
;extension=php_zip.dll
extension=php_pdo.dll
extension=php_pdo_sqlite.dll
;extension=php_pdo_firebird.dll
;extension=php_pdo_mssql.dll
;extension=php_pdo_mysql.dll
;extension=php_pdo_oci.dll
;extension=php_pdo_oci8.dll
;extension=php_pdo_odbc.dll
;extension=php_pdo_pgsql.dll
```

These two extensions will not be commented out. At loadtime, these extensions will now be included in the PHP environment, provided that the file php.ini is set..



phpinfo() - Windows Internet Explorer

http://localhost:8081/info.php

File Edit View Favorites Tools Help

Google G Go 87 blocked Check AutoL

Sample form to take ... WAMP5 is now Wam... phpinfo()

mbstring.strict_detection	Off
mbstring.substitute_character	no value

mysql

MySQL Support		enabled
Active Persistent Links	0	
Active Links	0	
Client API version	5.0.45	

Directive	Local Value	Master Value
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	no value	no value
mysql.default_socket	no value	no value
mysql.default_user	no value	no value
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

mysqli

Mysqli Support		enabled
Client API library version	5.0.45	
Client API header version	5.0.45	
MYSQLI_SOCKET	/tmp/mysql.sock	

Local intranet 100%

PHP should be configured for MySQL. You can verify that the php.ini file was properly read and the MySQL extensions are loaded by running the info.php script and looking for these entries.

PHP and Database Connectivity (cont.)

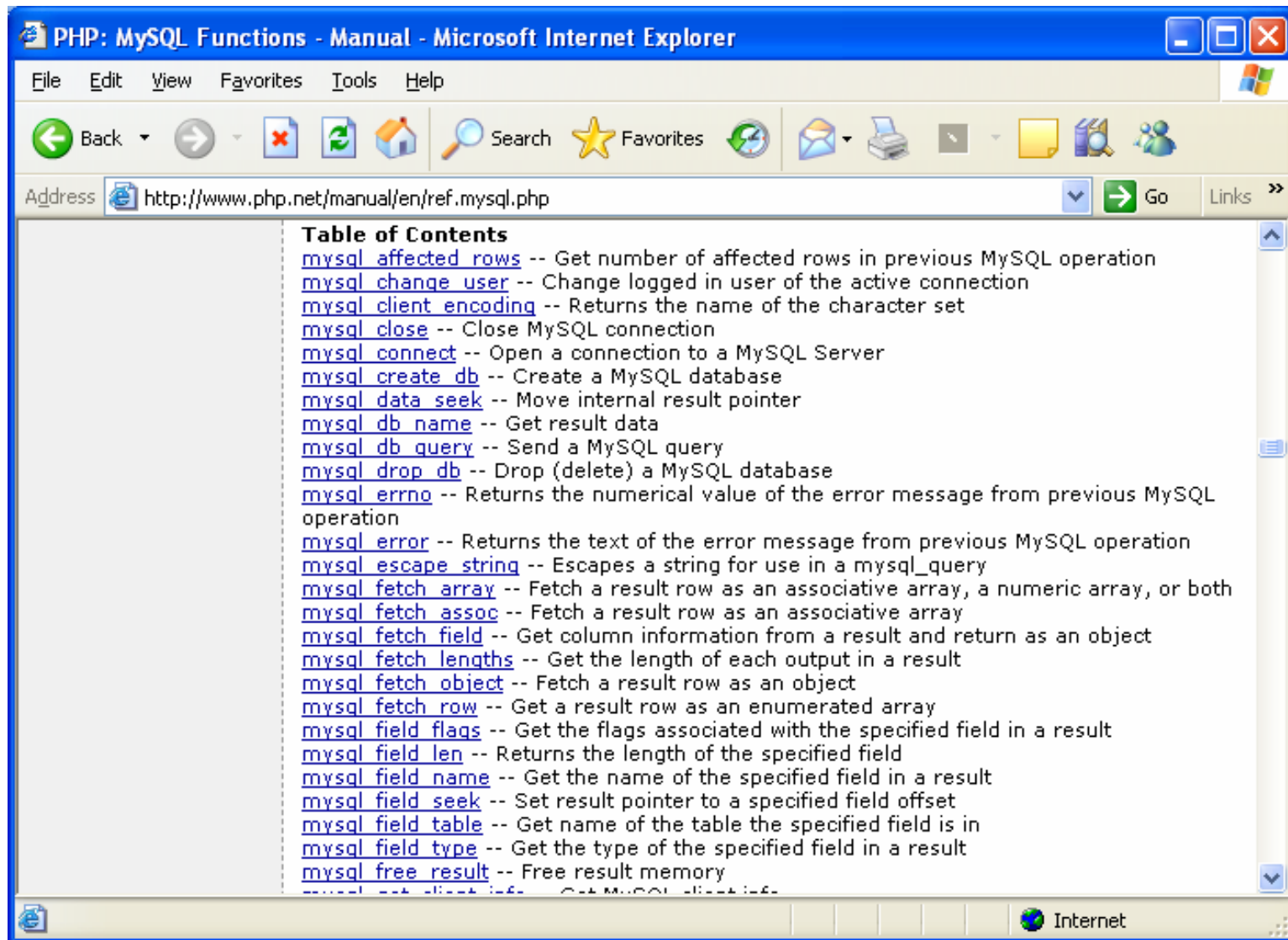
- PHP contains a fairly extensive set of commands that can be used to access and manipulate MySQL databases.
- A very brief listing of some of these commands appears on the next page.
- For a complete listing see:

<http://us2.php.net/manual/en/print/ref.mysql.php>.

<http://us2.php.net/manual/en/print/ref.mysql.php>.



Portion of `mysql.dll` Extension



PHP: MySQL Functions - Manual - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <http://www.php.net/manual/en/ref.mysql.php> Go Links

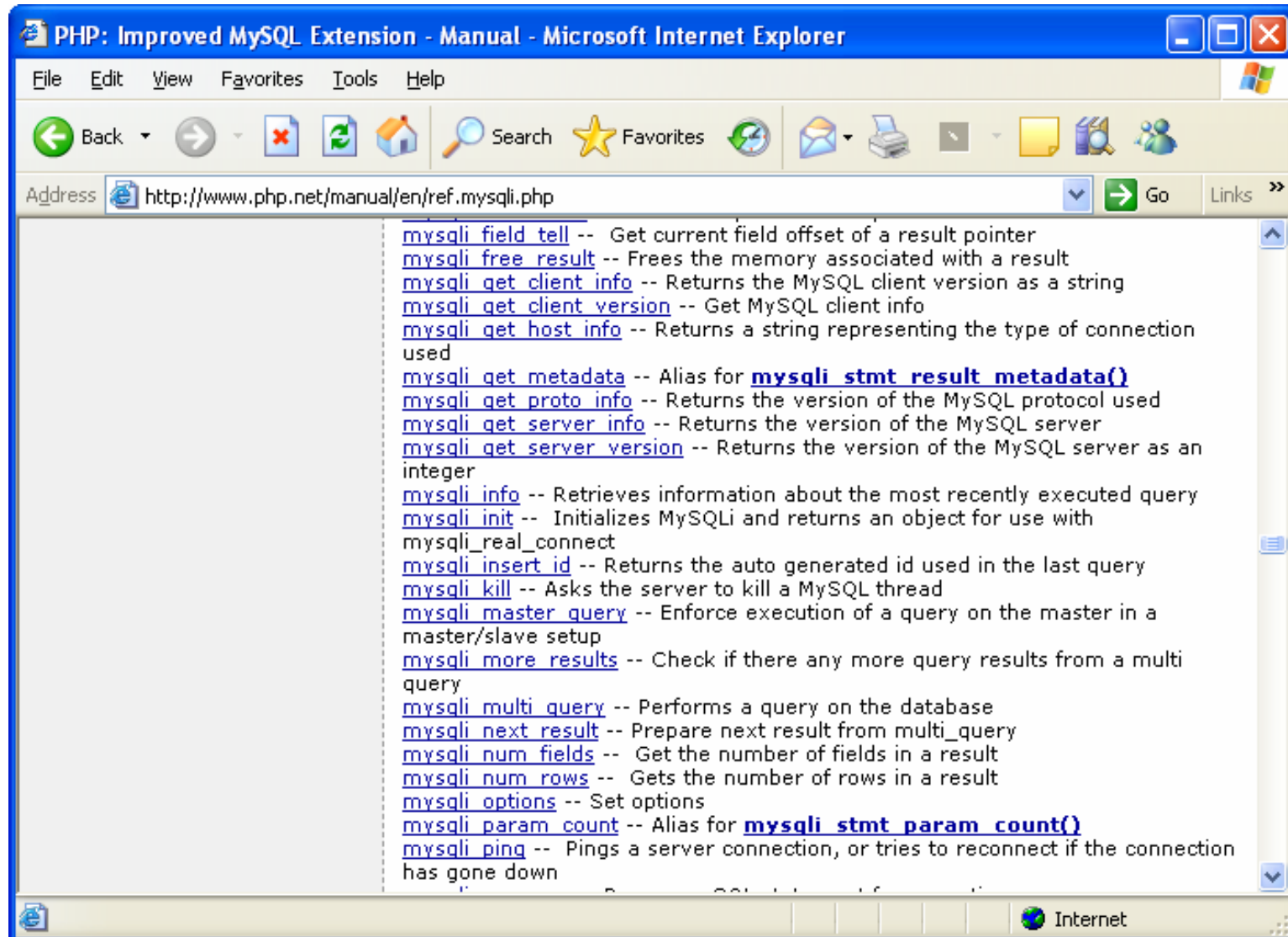
Table of Contents

- [mysql_affected_rows](#) -- Get number of affected rows in previous MySQL operation
- [mysql_change_user](#) -- Change logged in user of the active connection
- [mysql_client_encoding](#) -- Returns the name of the character set
- [mysql_close](#) -- Close MySQL connection
- [mysql_connect](#) -- Open a connection to a MySQL Server
- [mysql_create_db](#) -- Create a MySQL database
- [mysql_data_seek](#) -- Move internal result pointer
- [mysql_db_name](#) -- Get result data
- [mysql_db_query](#) -- Send a MySQL query
- [mysql_drop_db](#) -- Drop (delete) a MySQL database
- [mysql_errno](#) -- Returns the numerical value of the error message from previous MySQL operation
- [mysql_error](#) -- Returns the text of the error message from previous MySQL operation
- [mysql_escape_string](#) -- Escapes a string for use in a `mysql_query`
- [mysql_fetch_array](#) -- Fetch a result row as an associative array, a numeric array, or both
- [mysql_fetch_assoc](#) -- Fetch a result row as an associative array
- [mysql_fetch_field](#) -- Get column information from a result and return as an object
- [mysql_fetch_lengths](#) -- Get the length of each output in a result
- [mysql_fetch_object](#) -- Fetch a result row as an object
- [mysql_fetch_row](#) -- Get a result row as an enumerated array
- [mysql_field_flags](#) -- Get the flags associated with the specified field in a result
- [mysql_field_len](#) -- Returns the length of the specified field
- [mysql_field_name](#) -- Get the name of the specified field in a result
- [mysql_field_seek](#) -- Set result pointer to a specified field offset
- [mysql_field_table](#) -- Get name of the table the specified field is in
- [mysql_field_type](#) -- Get the type of the specified field in a result
- [mysql_free_result](#) -- Free result memory
- [mysql_get_client_info](#) -- Get MySQL client info

Internet



Portion of `mysqli.dll` Extension



The screenshot shows a Microsoft Internet Explorer browser window with the title "PHP: Improved MySQL Extension - Manual - Microsoft Internet Explorer". The address bar displays the URL "http://www.php.net/manual/en/ref.mysqli.php". The main content area lists various `mysqli` functions with their descriptions:

- [mysqli_field_tell](#) -- Get current field offset of a result pointer
- [mysqli_free_result](#) -- Frees the memory associated with a result
- [mysqli_get_client_info](#) -- Returns the MySQL client version as a string
- [mysqli_get_client_version](#) -- Get MySQL client info
- [mysqli_get_host_info](#) -- Returns a string representing the type of connection used
- [mysqli_get_metadata](#) -- Alias for [mysqli_stmt_result_metadata\(\)](#)
- [mysqli_get_proto_info](#) -- Returns the version of the MySQL protocol used
- [mysqli_get_server_info](#) -- Returns the version of the MySQL server
- [mysqli_get_server_version](#) -- Returns the version of the MySQL server as an integer
- [mysqli_info](#) -- Retrieves information about the most recently executed query
- [mysqli_init](#) -- Initializes MySQLi and returns an object for use with `mysqli_real_connect`
- [mysqli_insert_id](#) -- Returns the auto generated id used in the last query
- [mysqli_kill](#) -- Asks the server to kill a MySQL thread
- [mysqli_master_query](#) -- Enforce execution of a query on the master in a master/slave setup
- [mysqli_more_results](#) -- Check if there any more query results from a multi query
- [mysqli_multi_query](#) -- Performs a query on the database
- [mysqli_next_result](#) -- Prepare next result from `multi_query`
- [mysqli_num_fields](#) -- Get the number of fields in a result
- [mysqli_num_rows](#) -- Gets the number of rows in a result
- [mysqli_options](#) -- Set options
- [mysqli_param_count](#) -- Alias for [mysqli_stmt_param_count\(\)](#)
- [mysqli_ping](#) -- Pings a server connection, or tries to reconnect if the connection has gone down



PHP and Database Connectivity (cont.)

- Now that you have PHP set to accept MySQL extensions, let's connect to the bike database that we used for examples with Java servlets and JSPs.
- The following example is a simple database connection process in PHP where the client interacts with the database from an XHTML form that simply asks them to select which attributes from the bikes table that they would like to display. This is done through the `data.html` file.
- When the client clicks the submit query button, the `database.php` script executes by connecting to the database, posting the query, retrieving the results, and displaying them to the client.



data.html
Client side

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- data.html -->
<!-- Querying a MySQL Database From a PHP Script -->

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>   <title>Sample Database Query From PHP</title>  </head>
  <body style = "background-color: #545454" background=image1.jpg >
    <h2 style = "font-family: arial color: blue"> Querying a MySQL database from a PHP Script. </h2>
    <form method = "post" action = "database.php">
      <p>Select a field to display:
        <!-- add a select box containing options for SELECT query -->
        <select name = "select">
          <option selected = "selected">*</option>
          <option>bikename</option>
          <option>size</option>
          <option>color</option>
          <option>cost</option>
          <option>purchased</option>
          <option>mileage</option>
        </select>
      </p>
      <input type = "submit" value = "Send Query" style = "background-color: blue;
        color: yellow; font-weight: bold" />
    </form>
  </body> </html>
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- database.php -->
<!-- Program to query a database and send results to the client. -->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>   <title>Database Search Results</title>  </head>

  <body style = "font-family: arial, sans-serif"
  style = "background-color: #4A766E" background=image1.jpg link=blue vlink=blue>
  <?php
```

```
extract( $_POST );
```

```
// build SELECT query
$query = "SELECT " . $select . " FROM bikes";
```

```
// Connect to MySQL
if ( !( $database = mysqli_connect( "localhost",
  "root", "root", bikedb ) ) )
  die( "Could not connect to database" );
```

Default query is to select the attributes chosen by the client for use in a SELECT query.

Connect to MySQL database. URL, username, password, and database all specified.



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

```
<h3 style = "color: blue">
```

```
Database Search Results</h3>
```

```
<table border = "1" cellpadding = "3" cellspacing = "3"
style = "background-color: #00FFFF"> <!-- ADD8E6 -->
```

```
<?php
```

```
    // fetch meta-data
```

```
    $metadata = mysqli_fetch_fields( $result);
```

```
    print("<tr>");
```

```
    for ($i=0; $i<count($metadata); $i++){
```

```
        print("<td>");
```

```
        printf("%s", $metadata[$i]->name);
```

```
        print("</td>");
```

```
    }
```

```
    print("</tr>");
```

Get metadata for
the query

Display metadata in the
top row of the table



```
// 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>" );
}
mysql_close( $database );
?>
</table>
<br />Your search yielded <strong>
    <?php print( "$counter" ) ?> results.<br /><br /></strong>
<h5>Please email comments to
    <a href = "mailto:markl@cs.ucf.edu">
        markl@cs.ucf.edu
    </a>
</h5>
</body></html>
```



Execution of data.html – Client side

Sample Database Query From PHP - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8081/data.html> Go Links

Querying a MySQL database from a PHP Script.

Select a field to display:

Send Query

- * (selected)
- bikename
- size
- color
- cost
- purchased
- mileage

Execution of data.html (client side of the application) showing the drop-down menu for the client to select the attributes for the query.

When the selection is made and the **Send Query** button is clicked the results on the following page will be displayed.



Execution of database.php – Server side

Database Search Results

bikename	size	color	cost	purchased	mileage
Colnago Dream Rabobank	60	blue/orange	5500	2002-07-27	4300
Bianchi Evolution 3	58	celeste	4800	2003-11-16	2000
Eddy Merckx Molteni	58	orange	5100	2004-08-12	0
Eddy Merckx Domo	58	blue/black	5300	2004-02-02	0
Battaglin Carrera	60	red/white	4000	2001-03-14	11200
Gianni Motta Personal	59	red/green	4400	2000-05-01	8700
Gios Torino Super	60	blue	2000	1998-11-08	9000
Schwinn Paramount P14	60	blue	1800	1992-03-01	200
Bianchi Corse Evo 4	58	celeste	5700	2004-12-22	300
Colnago Superissimo	59	red	3800	1996-03-01	13000
Liquigas Bianchi FG Lite	59	celeste/blue/gr	5600	2005-06-27	0

Results of query **SELECT * FROM bikes**. Display indicates that 10 rows were included in the result.



Cookies

- A **cookie** is a text file that a Web site stores on a client's computer to maintain information about the client during and between browsing sessions.
- A Web site can store a cookie on a client's computer to record user preferences and other information that the Web site can retrieve during the client's subsequent visits. For example, many Web sites use cookies to store client's zipcodes. The Web site can retrieve the zipcode from the cookie and provide weather reports and news updates tailored to the user's region.
- Web sites also use cookies to track information about client activity. Analysis of information collected via cookies can reveal the popularity of Web sites or products.



Cookies (cont.)

- Marketers use cookies to determine the effectiveness of advertising campaigns.
- Web sites store cookies on users' hard drives, which raises issues regarding security and privacy. Web sites should not store critical information, such as credit-card numbers or passwords, in cookies, because cookies are just text files that anyone can read.
- Several cookie features address security and privacy concerns. A server can access only the cookies that it has placed on the client.
- A cookies has an expiration date, after which the Web browser deletes it.



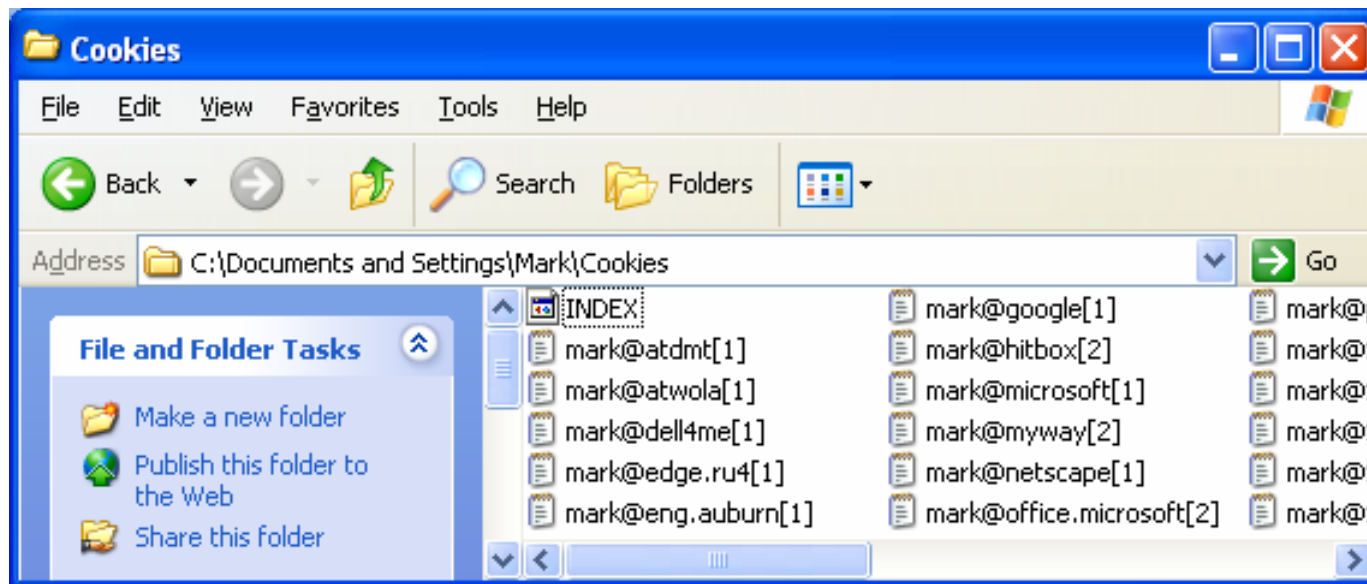
Cookies (cont.)

- Users who are concerned about the privacy and security implications of cookies can disable them in their Web browsers. However, the disabling of cookies can make it impossible for the user to interact with Web sites that rely on cookies to function properly.
- Information stored in the cookie is sent to the Web server from which it originated whenever the user requests a Web page from that particular server. The Web server can send the client XHTML output that reflects the preferences or information that is stored in the cookie.
- The location of the cookie file varies from browser to browser. Internet Explorer places cookies in the Cookies directory located at `C:\Documents and Settings\...\Cookies`



Cookies (cont.)

- After a cookie is created, a text file is added to this directory. While the name of the file will vary from user to user a typical example is shown below.



- The contents of a cookie are shown on page 43.



Cookies (cont.)

- Now let's create the code necessary to create our own cookie.
- In this example, a PHP script is invoked from a client-side HTML document. The HTML document creates a form for the user to enter the information that will be stored in the cookie. (Often the information that is stored in a cookie will be extracted from several different areas and may involved tracking the client's actions at the Web site.)
- Once the user has entered their information, when they click the Write Cookie button, the `cookies.php` script executes.
- The XHTML document and the PHP script are shown on the next pages. The XHTML document `cookies.html` is on page 36 and the PHP script `cookies.php` appears on page 37.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!-- cookies.html -->
```

```
<!-- Writing a Cookie -->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title>Writing a cookie to the client computer</title> </head>
```

```
  <body style = "font-family: arial, sans-serif;
  background-color: #856363" background=image1.jpg>
  <h2>Click Write Cookie to save your cookie data.</h2>
```

```
  <form method = "post" action = "cookies.php" style = "font-size: 10pt"
    background-color: #856363">
    <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 />
    <p>
      <input type = "submit" value = "Write Cookie" style = "background-color: #0000FF;
        color: yellow; font-weight: bold" /></p>
  </form>
</body> </html>
```



```

<?php
// cookies.php
// Program to write a cookie to a client's machine
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 );
?>

```

Function setcookie sets the cookies to the values passed from the cookies.html form. Function setcookie prints XHTML header information and therefore it needs to be called before any other XHTML (including comments) is printed.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

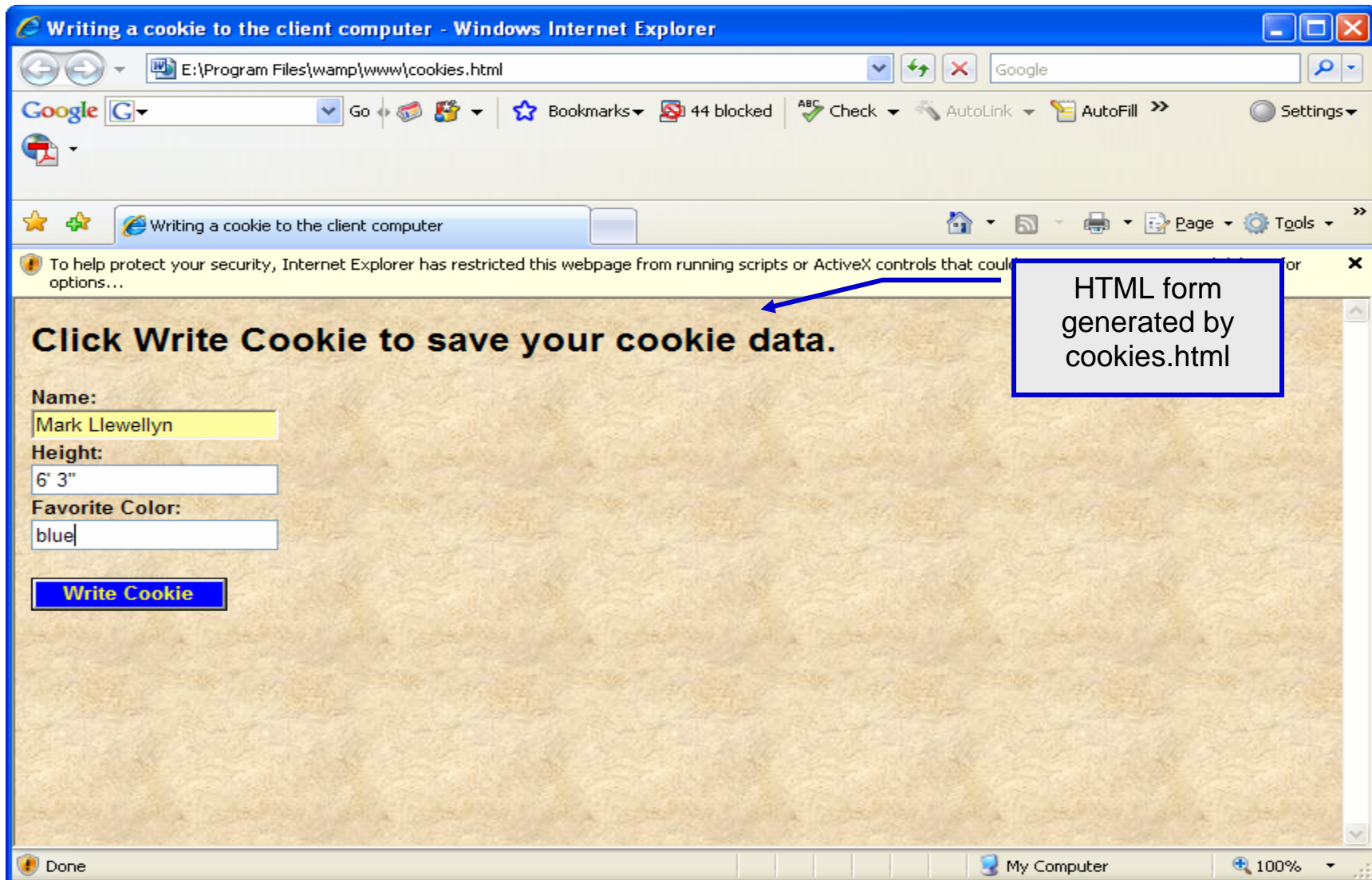
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title>Cookie Saved</title> </head>
<body style = "font-family: arial, sans-serif", background=image1.jpg">
<p><b>The cookie has been set with the following data:</b></p>
<!-- print each form field's value -->
<br /><span style = "color: blue">Name:</span>
<?php print( $NAME ) ?><br />
<span style = "color: blue">Height:</span>
<?php print( $HEIGHT ) ?><br />
<span style = "color: blue">Favorite Color:</span>
<span style = "color: <?php print( "$COLOR\">$COLOR" ) ?>
</span><br />
<p>Click <a href = "readCookies.php">here</a> to read the saved cookie.</p>
</body> </html>

```

The third argument to setcookie is optional and indicates the expiration date of the cookie. In this case it is set to expire 5 days from the current time. Function time returns the current time and then we add to this the number of seconds after which the cookie is to expire.



Cookies (cont.)

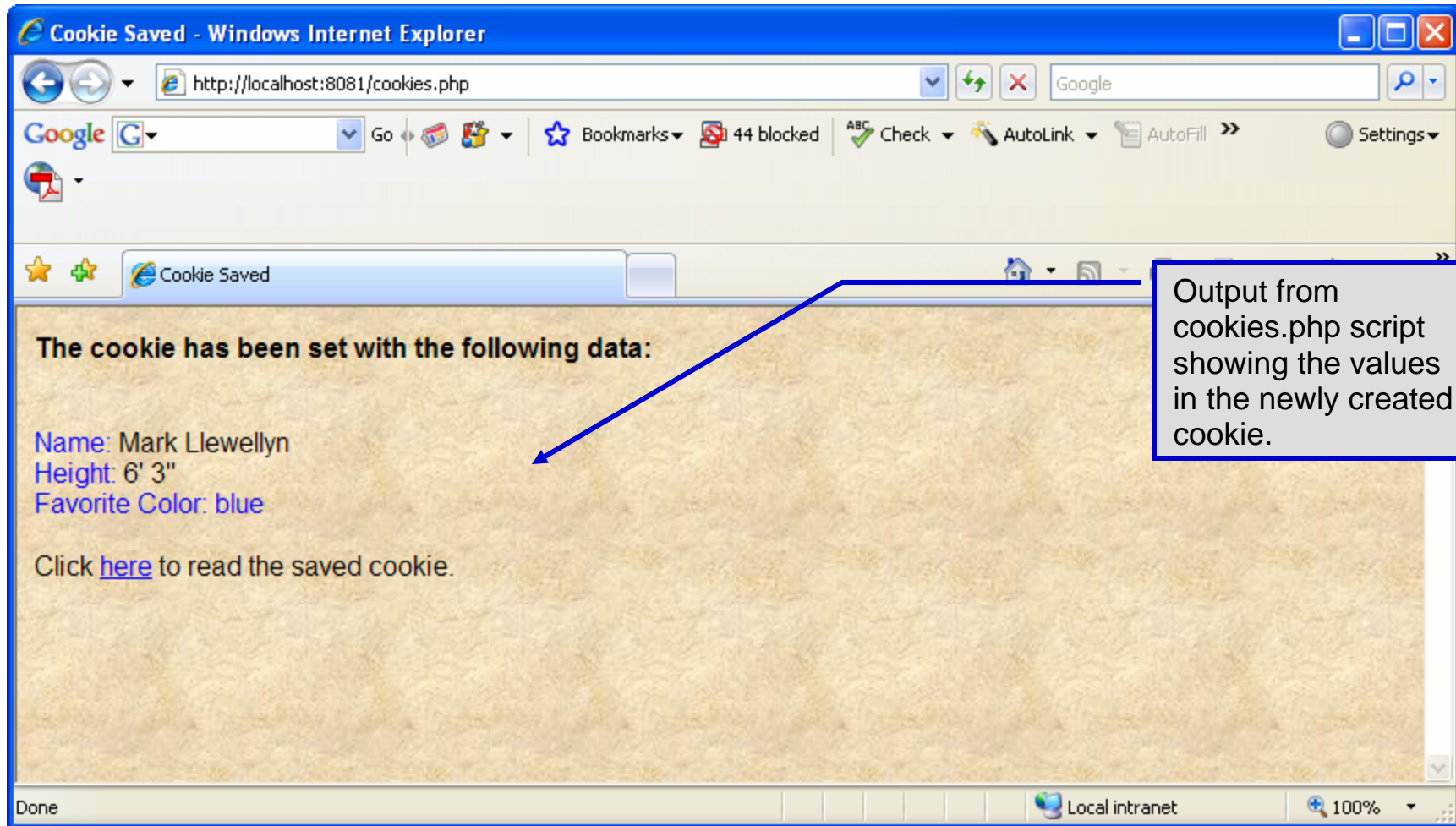


The screenshot shows a Windows Internet Explorer browser window titled "Writing a cookie to the client computer - Windows Internet Explorer". The address bar shows the file path "E:\Program Files\wamp\www\cookies.html". The browser interface includes a search bar with "Google" and various toolbars. A security warning is visible at the top of the page content. The main content area features a large heading "Click Write Cookie to save your cookie data." followed by a form with three input fields: "Name:" containing "Mark Llewellyn", "Height:" containing "6' 3\"", and "Favorite Color:" containing "blue". A blue "Write Cookie" button is positioned below the form. A blue arrow points from a text box to the "Write Cookie" button.

HTML form generated by cookies.html



Cookies (cont.)



Cookies (cont.)

- Once the cookie has been created, the cookies.php script gives the user the chance to view the newly created cookie by invoking the readCookies.php script from within the cookies.php script by clicking on the link.
- The readCookies.php script code is illustrated on the next page followed by the output from the execution of this PHP script.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!-- readCookies.php -->
<!-- Program to read cookies from the client's computer -->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>Read Cookies</title></head>
```

```
<body style = "font-family: arial, sans-serif" background=image1.jpg>
<p>
<strong> The following data is saved in a cookie on your computer.
</strong>
</p>
<table border = "5" cellspacing = "0" cellpadding = "10">
```

```
<?php
// iterate through array $_COOKIE and print
// name and value of each cookie
foreach ( $_COOKIE as $key => $value )
print( "<tr>
<td bgcolor=\"#F0E68C\">$key</td>
<td bgcolor=\"#FFA500\">$value</td>
</tr>" );
?>
</table>
</body> </html>
```

Superglobal array
holding cookie.



Cookies (cont.)

Read Cookies - Windows Internet Explorer

http://localhost:8081/readCookies.php

Google

Go

Bookmarks

44 blocked

Check

AutoLink

AutoFill

Settings

Check spelling of text you type on any web form in English

Read Cookies

Tools

The following data is saved in a cookie on your computer.

Name	Mark Llewellyn
Height	6' 3"
Color	blue

Done

Local intranet

100%

Output from the readCookies.php script.



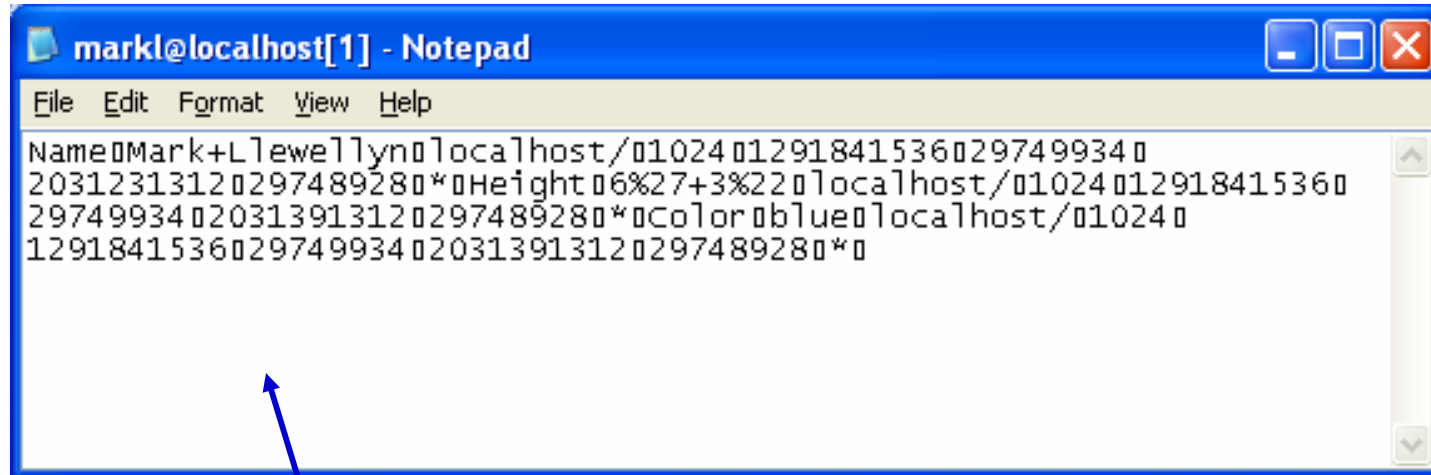
Cookies (cont.)

Windows Internet Explorer window showing the output of a PHP script in an Apache environment. The output is a table of variables and their values. A blue arrow points to the `HTTP_COOKIE` value, which is highlighted in a blue box. A callout box points to this value with the text: "Contents of the cookie stored on the client machine."

Variable	Value
HTTP_ACCEPT	*/*
HTTP_ACCEPT_LANGUAGE	en-us
HTTP_UA_CPU	x86
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.0.3705; .NET CLR 1.1.4322; Media Center PC 4.0)
HTTP_HOST	localhost
HTTP_CONNECTION	Keep-Alive
HTTP_COOKIE	Name=Mark+Llewellyn; Height=6+ft+3+in; Color=blue
PATH	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;c:\Python22;C:\Program Files\QuickTime\QTSystem\
SystemRoot	C:\WINDOWS
COMSPEC	C:\WINDOWS\system32\cmd.exe
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
WINDIR	C:\WINDOWS
SERVER_SIGNATURE	no value
SERVER_SOFTWARE	Apache/2.2.8 (Win32) PHP/5.2.5
SERVER_NAME	localhost
SERVER_ADDR	127.0.0.1
SERVER_PORT	80
REMOTE_ADDR	127.0.0.1



Cookies (cont.)



A screenshot of a Notepad window titled "markl@localhost[1] - Notepad". The window contains the following text:

```
Name=Mark+Llewellyn;localhost/0102401291841536029749934020312313120297489280*0Height=6%27+3%22;localhost/0102401291841536029749934020313913120297489280*0Color=blue;localhost/0102401291841536029749934020313913120297489280*
```

Actual text file holding cookie data for the cookie that was created in this example.



Dynamic Content in PHP

- Of all the strengths PHP exhibits as a server-side scripting language, perhaps its greatest strength lies in its ability to dynamically change XHTML output based on user input.
- In this final section of notes, we'll build on the examples we've constructed in the previous two sets of notes by combining `form.html` and `form.php` into one dynamic PHP document named `dynamicForm2.php`.
- We'll add error checking to the user input fields and inform the user of invalid entries on the form itself, rather than on an error page. If an error exists, the script maintains the previously submitted values in each form element.
- Finally, after the form has been successfully completed, we'll store the input from the user in a MySQL database.



Sample form to take user input in XHTML - Windows Internet Explorer

http://localhost:8081/dynamicform2.php

File Edit View Favorites Tools Help

Google G Go 44 blocked Check AutoLink AutoFill Settings

Sample form to take user input in XHTML

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name

Last Name

Email

Phone

Must be in the form (555)555-5555

Publications

Which magazine would you like information about?

Velo-News

Operating System

Which operating system are you currently using?

Windows XP Windows 2000 Windows 98
 Linux Other

Basically, the same registration form that was used in a previous example.

Done Local intranet 100%



Sample form to take user input in XHTML - Windows Internet Explorer

http://localhost:8081/dynamicform2.php

File Edit View Favorites Tools Help

Google G Go 44 blocked Check AutoLink AutoFill Settings

Sample form to take user input in XHTML

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name

Last Name

Email

Phone

Must be in the form (555)555-5555

Publications

Which magazine would you like information about?

Operating System

Which operating system are you currently using?

Windows XP Windows 2000 Windows 98

Linux Other

User fills in the form and clicks the Register button.

Local intranet 100%



Sample form to take user input in XHTML - Windows Internet Explorer

http://localhost:8081/dynamicForm2.php

File Edit View Favorites Tools Help

Google G Go Bookmarks 44 blocked Check AutoLink AutoFill Settings

Sample form to take user input in XHTML

Hi **Mark**. Thank you for completing the survey.
You have been added to the **Pro Cycling** mailing list.

The following information has been saved in our database:

Name	Email	Phone	OS
Mark Llewellyn	markl@cs.ucf.edu	(407)823-2790	Windows XP

[Click here to view entire database.](#)

Done Local intranet 100%

Screen the user sees after clicking the **Register** button.



Database Search Results - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <http://localhost:8081/formDatabase2.php>

Screen the user sees after clicking to see the entire database.

Mailing List Contacts

ID	Last Name	First Name	E-mail Address	Phone Number	Magazine	Operating System
1	Schumacher	Ralf	ralf@toyota.com	(222)222-222	Cycling Weekly	Other
2	Thurau	Didi	agirl.de	(456)123-3345	Mirror du Cylisme	Windows 98
3	Jolie	Angelina	aj@hollywood.com	(123)456-7890	Velo-news	Windows XP
4	Schumacher	Michael	michael@ferrari.it	(407)823-2790	RadSport	Windows 2000
5	Campbell	Kristi	im_not_sure	(345)987-6545	Cycle Sport	Windows 2000
6	Einstein	Albert	relative	(111)111-1111	Cycling Weekly	Other
7	Llewellyn	Mark	markl@cs.ucf.edu	(407)823-2790	Pro Cycling	Windows XP

Done Local intranet



The screenshot shows a Windows Internet Explorer browser window titled "Sample form to take user input in XHTML - Windows Internet Explorer". The address bar shows "http://localhost:8081/dynamicForm2.php". The browser interface includes a menu bar (File, Edit, View, Favorites, Tools, Help), a search bar with "Google" and "Go" buttons, and a toolbar with various icons. The main content area has a light brown background and contains the following text and form elements:

- Title:** "This is a sample registration form."
- Instructions:** "Please fill in all fields and click Register. Fields with * need to be filled in properly."
- User Information Section:**
 - Section header: "User Information" (with a dropdown arrow)
 - Instruction: "Please fill out the fields below."
 - Form fields:
 - First Name:** An empty text input field with a red asterisk (*) to its right.
 - Last Name:** A text input field containing "I-forgot".
 - Email:** A text input field containing "deadl-letter".
 - Phone:** A text input field containing "(555)555-5555".
 - Validation message: "Must be in the form (555)555-5555"
- Publications Section:**
 - Section header: "Publications" (with a dropdown arrow)
 - Instruction: "Which magazine would you like information about?"
 - Form element: A dropdown menu currently showing "Cycling Weekly".
- Operating System Section:**
 - Section header: "Operating System" (with a dropdown arrow)
 - Instruction: "Which operating system are you currently using?"
 - Form elements: Radio buttons for "Windows XP" (selected), "Windows 2000", "Windows 98", "Linux", and "Other".
- Form Control:** A "Register" button at the bottom of the form.

A callout box on the right side of the browser window contains the following text:

Dynamic nature of the PHP form is illustrated when the user fails to enter proper information into the form. In this case, the user forgot to enter their first name. Error checking is in place on each user input location and the page is dynamically updated to reflect the error processing and correction capabilities. The database will not be updated until the user has correctly filled in all required fields.

At the bottom of the browser window, the status bar shows "Done", "Local intranet", and "100%".



MySQL Query Browser - root@localhost:3306 / mailinglist

File Edit View Query Script Tools Help

Go back Next Refresh

```
SELECT * FROM `contacts` c
```

Execute Stop

Transaction Explain Compare

SELECT FROM WHERE GROUP HAVING ORDER SET CREATE

Resultset 1

ID	LastName	FirstName	Email	Phone	Magazine	OS
1	Schumacher	Ralf	ralf@toyota.com	(222)222-222	Cycling Weekly	Other
2	Thurau	Didi	agirl.de	(456)123-3345	Mirror du Cylisme	Windows 9
3	Jolie	Angelina	aj@hollywood.com	(123)456-7890	Velo-news	Windows X
4	Schumacher	Michael	michael@ferrari.it	(407)823-2790	RadSport	Windows 2
5	Campbell	Kristi	im_not_sure	(345)987-6545	Cycle Sport	Windows 2
6	Einstein	Albert	relative	(111)111-1111	Cycling Weekly	Other
7	Llewellyn	Mark	markl@cs.ucf.edu	(407)823-2790	Pro Cycling	Windows X

Schemata Bookmarks History

- bikedb
- book_sc
- coloursurvey
- guestbook
 - guests
- imagedb
- mailinglist
 - contacts

Syntax Functions Params Trx

- Data Manipulation
- Data Definition
- MySQL Utility
- Transactional and Locking

7 rows fetched in 0.0113s (0.0006s) Edit Apply Changes Discard

1: 1

Screen shot from MySQL of the contacts relation after the inclusion of several users. Note that the values in the table are the same as those returned to the PHP document in the previous slide.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- dynamicForm2.php      -->

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title>Sample form to take user input in XHTML</title>
  </head>
  <body style = "font-family: arial, sans-serif; background-color: #856363"
background=background.jpg>
  <?php
    extract ( $_POST );
    $iserror = false;
    // array of magazine titles
    $maglist = array( "Velo-News",
      "Cycling Weekly",
      "Pro Cycling",
      "Cycle Sport",
      "RadSport",
      "Mirror du Cyclisme" );
    // array of possible operating systems
    $systemlist = array( "Windows XP",
      "Windows 2000",
      "Windows 98",
      "Linux",
      "Other");
```



```
// array of name and alt values for the text input fields
$inputlist = array( "fname" => "First Name",
    "lname" => "Last Name",
    "email" => "Email",
    "phone" => "Phone" );
if ( isset ( $submit ) ) {
    if ( $fname == "" ) {
        $formerrors[ "fnameerror" ] = true;
        $iserror = true;
    }
    if ( $lname == "" ) {
        $formerrors[ "lnameerror" ] = true;
        $iserror = true;
    }
    if ( $email == "" ) {
        $formerrors[ "emailerror" ] = true;
        $iserror = true;
    }
    if ( !ereg( "^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$", $phone ) ) {
        $formerrors[ "phoneerror" ] = true;
        $iserror = true;
    }
    if ( !$iserror ) {
        // build INSERT query
        $query = "INSERT INTO contacts " .
            "(ID, LastName, FirstName, Email, Phone, Magazine, OS ) " .
            "VALUES (null, '$lname', '$fname', '$email', " .
            "\" . \"\" . quotemeta( $phone ) . \"', '$mag', '$os' )";
    }
}
```



```
// Connect to MySQL
if ( !( $database = mysql_connect( "localhost",
    "root", "root" ) ) )
    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, $database ) ) ) {
    print( "Could not execute query! <br />" );
    die( mysql_error() );
}
print( "<p>Hi
    <span style = 'color: blue'> <strong>$fname</strong></span>.
    Thank you for completing the survey.<br />
    You have been added to the <span style = 'color: blue'>
    <strong>$mag</strong></span> mailing list.          </p>
    <strong>The following information has been saved in our database:</strong><br />

    <table border = '0' cellpadding = '0' cellspacing = '10'>
    <tr>
    <td bgcolor = '#ffffaa'>Name </td>
    <td bgcolor = '#ffffbb'>Email</td>
    <td bgcolor = '#ffffcc'>Phone</td>
    <td bgcolor = '#ffffdd'>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 style = 'font-size : 10pt; text-align: center'>
    <div style = 'font-size : 18pt'>
    <a href = 'formDatabase2.php'>
    Click here to view entire database.</a>
    </div>
</div></body></html>" );
die();
}
}
print( "<h1>This is a sample registration form.</h1>
Please fill in all fields and click Register." );
if ( $iserror ) {
    print( "<br /><span style = 'color : red'>
Fields with * need to be filled in properly.</span>" );
}
print( "<!-- post form data to dynamicForm2.php -->
<form method = 'post' action = 'dynamicForm2.php'>
<img src = 'images/user.gif' alt = 'User' /><br />
<span style = 'color: blue'>
Please fill out the fields below.<br />
</span>

```

Invoke PHP script to see contents of entire database if user clicks this link. Code begins on page 14.

The form created is self-submitting (i.e., it posts to itself). This is done by setting the action to dynamicForm2.php



```

<!-- 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 style = 'color : red'>*</span>" );
    print( "<br />" );
}
print( "<span style = 'font-size : 10pt'" );
if ( $formerrors[ "phoneerror" ] ) print( "; color : red" );
print( "'>Must be in the form (555)555-5555
    </span><br /><br />
    <img src = 'images/downloads.gif'
    alt = 'Publications' /><br />
    <span style = 'color: blue'>
    Which magazine would you like information about?
    </span><br />
    <!-- create drop-down list containing magazine names -->
    <select name = 'mag'>" );
foreach ( $maglist as $currmag ) {
    print( "<option" );
    if ( ( $currmag == $mag ) )
        print( " selected = 'true'" );
    print( ">$currmag</option>" );
}

```

The \$\$variable notation specifies variable variables. PHP permits the use of variable variables to allow developers to reference variables dynamically. The expression \$\$variable could also be written as \${\$variable} for added clarity.



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

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

$counter = 0;

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

    if ( $currssystem == $os ) print( "checked = 'checked'" );
    if ( iserror && $counter == 0 ) print( "checked = 'checked'" );

    print( " />$currssystem" );

    if ( $counter == 2 ) print( "<br />" );
    $counter++;
}

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



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!-- formDatabase2.php -->
```

```
<!-- Program to query a database and send results to the client. -->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title>Database Search Results</title> </head>
<body style = "font-family: arial, sans-serif"
style = "background-color: #F0E68C" background=image1.jpg>
<?php
extract( $_POST );
// build SELECT query
$query = "SELECT * FROM contacts";
// Connect to MySQL
if ( !( $database = mysqli_connect( "localhost", "root", "root", MailingList ) ) )
die( "Could not connect to database" );
// query MailingList database
if ( !( $result = mysqli_query( $database, $query ) ) ) {
print( "Could not execute query! <br />" );
die( mysqli_error() );
}
?>
<h3 style = "color: blue">
Mailing List Contacts</h3>
```



```
<table border = "1" cellpadding = "3" cellspacing = "2"
style = "background-color: #ADD8E6">
<tr>
  <td>ID</td>
  <td>Last Name</td>
  <td>First Name</td>
  <td>E-mail Address</td>
  <td>Phone Number</td>
  <td>Magazine</td>
  <td>Operating System</td>
</tr>
<?php
// fetch each record in result set
for ( $counter = 0;
  $row = mysqli_fetch_row( $result );
  $counter++ ){
  // build table to display results
  print( "<tr>" );
  foreach ( $row as $key => $value )
    print( "<td>$value</td>" );
  print( "</tr>" );
}
mysqli_close( $database );
?>

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



MySQL Table Editor

Table Name: contacts Database: mailinglist Comment: InnoDB free: 7168 kB

Columns and Indices Table Options Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
ID	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input checked="" type="checkbox"/> ZEROFILL	NULL	
LastName	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
FirstName	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
Email	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
Phone	VARCHAR(14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
Magazine	VARCHAR(60)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
OS	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	

Indices Foreign Keys Column Details

PRIMARY

Index Settings

Index Name: PRIMARY

Index Kind: PRIMARY

Index Type: BTREE

Index Columns (Use Drag'n'Drop)

ID

Apply Changes Discard Changes Close

Schema of the MailingList database table contacts required for the PHP database example to work. Script is available on the code page.



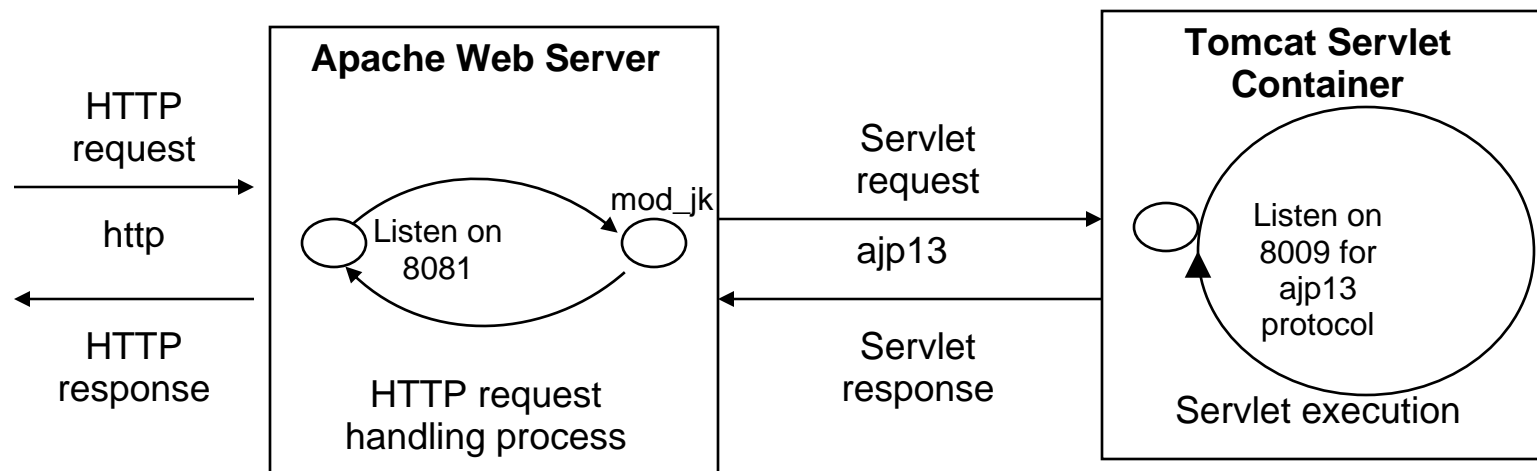
Connecting Apache To Tomcat

- Although it is possible for Tomcat to run standalone and serve HTTP requests directly (we did this for servlets and jsps), the Apache server does a much better job of handling tasks such as static content and SSL connections.
- For this reason, Tomcat is typically used alongside an Apache server. Unlike PHP which runs as a module inside the Apache process, a JVM is external and requires a mechanism to connect it to the web server.
- Tomcat inherited the Apache JServ protocol (AJP) from the JServ project. AJP is a protocol for connecting an external process to a servlet container. It is the responsibility of an Apache module, in this case `mod_jk`, to speak this protocol to the servlet container (Tomcat).



Connecting Apache To Tomcat (cont.)

- In this last section of notes, I'll show you how to integrate Apache and Tomcat into a single package.
- The ultimate set-up will resemble the figure shown below. Note that if you also would like Tomcat to run standalone HTTP requests, it will need to run on a different HTTP port than Apache. That's why I set-up Apache on port 8081 and Tomcat on port 8080.



Getting The Tomcat Connector

- The first step in Apache-Tomcat integration is to obtain the Tomcat connector from Apache.
- Follow the screen shots on the next few pages to obtain the `mod_jk` connector.
- The Windows based connector binary files has the name `mod_jk-apache-2.2.4.so`.




Apache Tomcat - Apache Tomcat - Windows Internet Explorer

http://tomcat.apache.org/


File Edit View Favorites Tools Help

Google G Go Bookmarks 44 blocked Check AutoLink AutoFill Settings

Apache Tomcat - Apache Tomcat



Apache Tomcat



The Apache Software Foundation

http://www.apache.org/

Apache Tomcat

- [Home](#)

Download

- [Which version?](#)
- [Tomcat 6.x](#)
- [Tomcat 5.x](#)
- [Tomcat 4.1](#)
- [Tomcat 3.3](#)
- [Tomcat Connectors](#)
- [Archives](#)

Documentation

Apache Tomcat is the... for the [Java Servlet](#) and... Pages specifications ar...

Apache Tomcat is developed in an open and participatory environment and released under the [Apache Software License](#). Apache Tomcat is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, [click here](#).

Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the [PoweredBy](#) wiki page.

http://www.apache.org/ Internet 100%

From the main Tomcat webpage, select Tomcat connectors from the download section. The current documentation is also available from this page.




Apache Tomcat - Tomcat Connectors (mod_jk) Downloads - Windows Internet Explorer

http://tomcat.apache.org/download-connectors.cgi


File Edit View Favorites Tools Help

Google G Go 44 blocked Check AutoLink AutoFill Settings

Apache Tomcat - Tomcat Connectors (mod_jk) Downl...



Apache Tomcat



The Apache Foundation

apache.org/

Search the Site Search Site

Apache Tomcat

- [Home](#)

Download

- [Which version?](#)
- [Tomcat 6.x](#)
- [Tomcat 5.x](#)
- [Tomcat 4.1](#)
- [Tomcat 3.3](#)
- [Tomcat Connectors](#)
- [Archives](#)

Documentation

Tomcat Connectors (mod_jk) Downloads

We recommend you use a mirror to download our release builds, but you **must** verify the integrity of the downloaded files using signatures downloaded from our main distribution directories. Recent releases (48 hours) may not yet be available from the mirrors.

You are currently using <http://www.axint.net/apache>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors:

The [KEYS](#) link links to the code signing keys used to sign the product. The [PGP](#) link downloads the OpenPGP compatible signature from our main site.

For more information concerning Tomcat Connectors (mod_jk), see the Tomcat

Done Internet 100%



Apache Tomcat - Tomcat Connectors (mod_jk) Downloads - Windows Internet Explorer

http://tomcat.apache.org/download-connectors.cgi

File Edit View Favorites Tools Help

Google G Go Bookmarks 44 blocked Check AutoLink AutoFill Settings

Apache Tomcat - Tomcat Connectors (mod_jk) Downl...

downloads the OpenPGP compatible signature

Documentation

- [Tomcat 6.0](#)
- [Tomcat 5.5](#)
- [Tomcat 5.0](#)
- [Tomcat 4.1](#)
- [Tomcat 3.3](#)
- [Tomcat Connectors](#)
- [Migration Guide](#)

Problems?

- [Security Reports](#)
- [Find help](#)
- [FAQ](#)
- [Mailing Lists](#)
- [Bug Database](#)
- [IRC](#)

Get Involved

- [Overview](#)
- [SVN Repositories](#)
- [Mailing Lists](#)

For more information concerning Tomcat Connectors (mod_jk) site.

KEYS

- **JK 1.2 (WARNING: Critical vulnerabilities in previous versions)**
 - o Source
 - [JK 1.2.21 Source Release tar.gz](#)
 - [\[pgp\]](#)
 - [JK 1.2.21 Source Release zip](#)
 - [\[pgp\]](#)
 - o [JK 1.2 Binary Releases](#)
- [browse download area](#)
- [archives...](#)
- The historical mod_jk2 could be found in the [archives](#)

Scroll down the page until you see the JK 1.1.21 binary releases

Done Internet 100%



Index of /apache/tomcat/tomcat-connectors/jk/binaries - Windows Internet Explorer











http://www.axint.net/apache/tomcat/tomcat-connectors/jk/binaries/

File Edit View Favorites Tools Help

Google G Go Bookmarks 44 blocked Check AutoLink AutoFill Settings

Index of /apache/tomcat/tomcat-connectors/jk/binaries

Index of /apache/tomcat/tomcat-connectors/jk/binaries

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory	24-Dec-2004 05:50	-	
 aix/	02-Mar-2007 17:06	-	
 freebsd/	02-Mar-2007 17:06	-	
 iseries/	02-Mar-2007 17:06	-	
 linux/	05-Mar-2007 04:03	-	
 macosx/	02-Mar-2007 17:06	-	
 netware/	05-Mar-2007 01:12	-	
 solaris/	05-Mar-2007 04:02	-	
 win32/	12-Apr-2007 23:48	-	
 win64/	12-Apr-2007 23:49	-	

Pick the directory of your choice for your machine. I'm using win32.

The Apache Jakarta Tomcat Connector

Done Internet 100%






Index of /apache/tomcat/tomcat-connectors/jk/binaries/win32 - Windows Internet Explorer

http://www.axint.net/apache/tomcat/tomcat-connectors/jk/binaries/win32/

File Edit View Favorites Tools Help

Google G Go Bookmarks 44 blocked Check AutoLink AutoFill Settings


Index of /apache/tomcat/tomcat-connectors/jk/binaries/win32

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory	04-Mar-2007 22:41	-	
 jk-1.2.21/	04-Mar-2007 22:38	-	
 jk-1.2.22/	13-Apr-2007 00:02	-	

The Apache Jakarta Tomcat Connector

- Current stable version is jk-1.2.21

Done Internet 100%



Make your choice here and select the version you want.















Index of /apache/tomcat/tomcat-connectors/jk/binaries/win32/jk-1.2.22 - Windows Internet Explorer

http://www.axint.net/apache/tomcat/tomcat-connectors/jk/binaries/win32/jk-1.2.22/

File Edit View Favorites Tools Help

Google G Go 44 blocked Check AutoLink AutoFill Settings

Index of /apache/tomcat/tomcat-connectors/jk/binaries/win32/jk-1.2.22

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory	12-Apr-2007 23:48	-	
 isapi_redirect.dll	12-Apr-2007 22:14	228k	
 isapi_redirect.dll.asc	12-Apr-2007 22:37	1k	
 jk-symbols.zip	12-Apr-2007 22:32	692k	
 jk-symbols.zip.asc	12-Apr-2007 22:37	1k	
 mod_jk-apache-1.3.37.so	12-Apr-2007 22:12	180k	
 mod_jk-apache-1.3.37.so.asc	12-Apr-2007 22:37	1k	
 mod_jk-apache-2.0.59.so	12-Apr-2007 22:11	188k	
 mod_jk-apache-2.0.59.so.asc	12-Apr-2007 22:37	1k	
 mod_jk-apache-2.2.4.so	12-Apr-2007 22:11	188k	
 mod_jk-apache-2.2.4.so.asc	12-Apr-2007 22:37	1k	
 nsapi_redirect.dll	12-Apr-2007 22:32	176k	

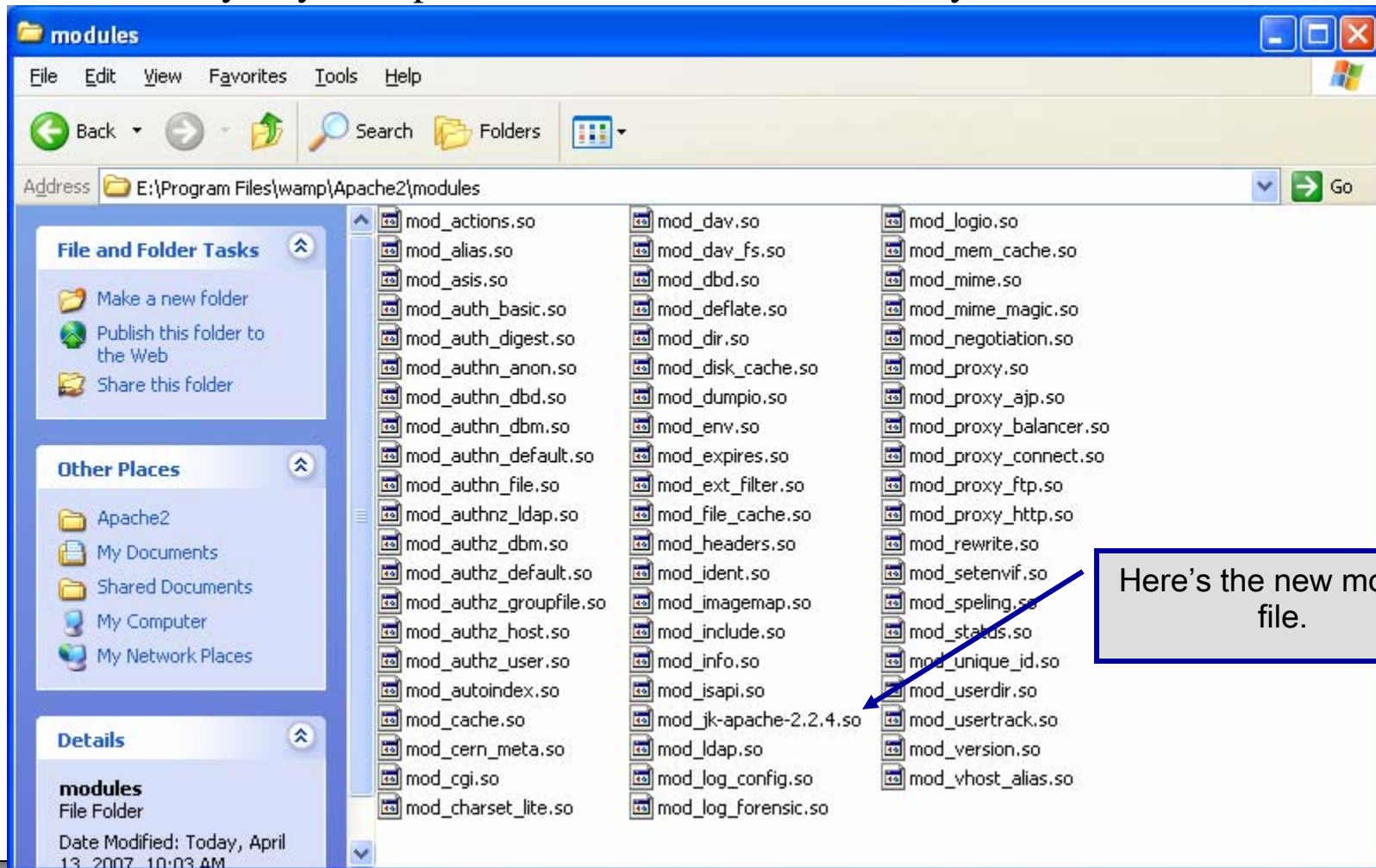
Internet 100%

Download this file and place it in the Wamp/Apache2/modules directory.



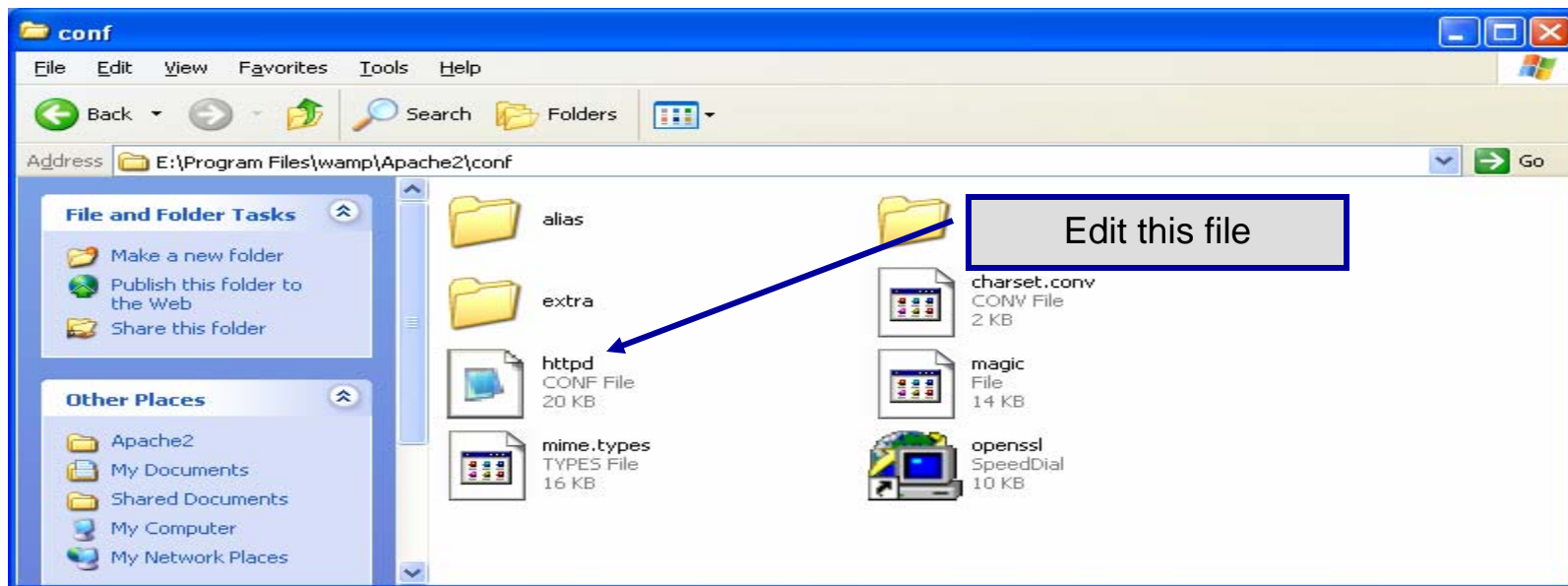
Installing The Tomcat Connector

- Once you've downloaded the Tomcat connector, put it in the modules sub-directory of your Apache installation's root directory.



Enabling The Tomcat Connector

- Once you've put the connector file in the `modules` sub-directory of your Apache installation's root directory, you are now ready to configure Apache to recognize and enable the module.
- To do this you'll need to edit your Apache configuration file. This is the same file you edited to enable PHP.
- This file is located in the `conf` subdirectory and is named `httpd`.



httpd - Notepad

File Edit Format View Help

```
### begin material for mod_jk connector
# Load mod_jk module
# Update this path to match your modules location
LoadModule jk_module "E:/Program Files/wamp/Apache2/modules/mod_jk-apache-2.2.4|.so"

# Declare the module for <IfModule directive> (remove this line on Apache 2.x)
#AddModule mod_jk.c

# where to find workers.properties
# Update this path to match your conf directory location (put workers.properties next to httpd.conf)
#JkWorkersFile "E:/Program Files/Apache Software Foundation/Tomcat 5.5.20/conf/workers.properties"

# where to put jk logs
# Update this path to match your logs directory location (put mod_jk.log next to access_log)
#JkLogFile "E:/Program Files/wamp/Apache2/logs/mod_jk.log"

# set the jk log level [debug/error/info]
#JkLogLevel info

# select the log format
#JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

# JkOptions indicate to send SSL KEY SIZE,
#JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat set the request format
#JkRequestLogFormat "%w %V %T"

# send everything for context /examples to worker named worker1 (ajp13)
#JkMount /examples/* worker1
JkMount /cop4610/* ajp13
JkMount /first-examples/* ajp13

### end material for mod_jk connector
```

Add all of this material to enable mod_jk

Add JKMount directives for any servlets and jsp's to go through Tomcat.



- The `JWorkerFile` directive (see previous page) refers to a separate file that configures the `ajp13` protocol communications parameters. An example called `workers.properties` is included with `mod_jk`. For a basic set-up, this default file will work fine.
- There are a couple of things you should verify however: (1) `workers.tomcat_home` should agree with the value you've already set for Tomcat called `CATALINA_HOME`, and (2) `worker.ajp13.port` must be the same as the one listed in the `Tomcat server.xml` file as shown on the next page.
- In more robust applications, additional editing of the `JWorkerFile` will be required. For example, if you have more than one installation of Tomcat on your machine, you'll need to adjust the `worker.ajp.port` parameter in `workers.properties` to make sure that `mod_jk` is connecting to the correct Tomcat installation as Tomcat installations will not be able to start up sharing port numbers.



```
workers - WordPad
File Edit View Insert Format Help
[Icons]
worker.ajp12.lbfactor=1

#
#----- DEFAULT ajp13 WORKER DEFINITION -----
#-----
#
#
# Defining a worker named ajp13 and of type ajp13
# Note that the name and the type do not have to match.
#
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
#
# Specifies the load balance factor when used with
# a load balancing worker.
# Note:
# ----> lbfactor must be > 0
# ----> Low lbfactor means less work done by the worker.
worker.ajp13.lbfactor=1

#
# Specify the size of the open connection cache.
#worker.ajp13.cachesize

#
#----- DEFAULT LOAD BALANCER WORKER DEFINITION -----
#-----
#
#
For Help, press F1
```

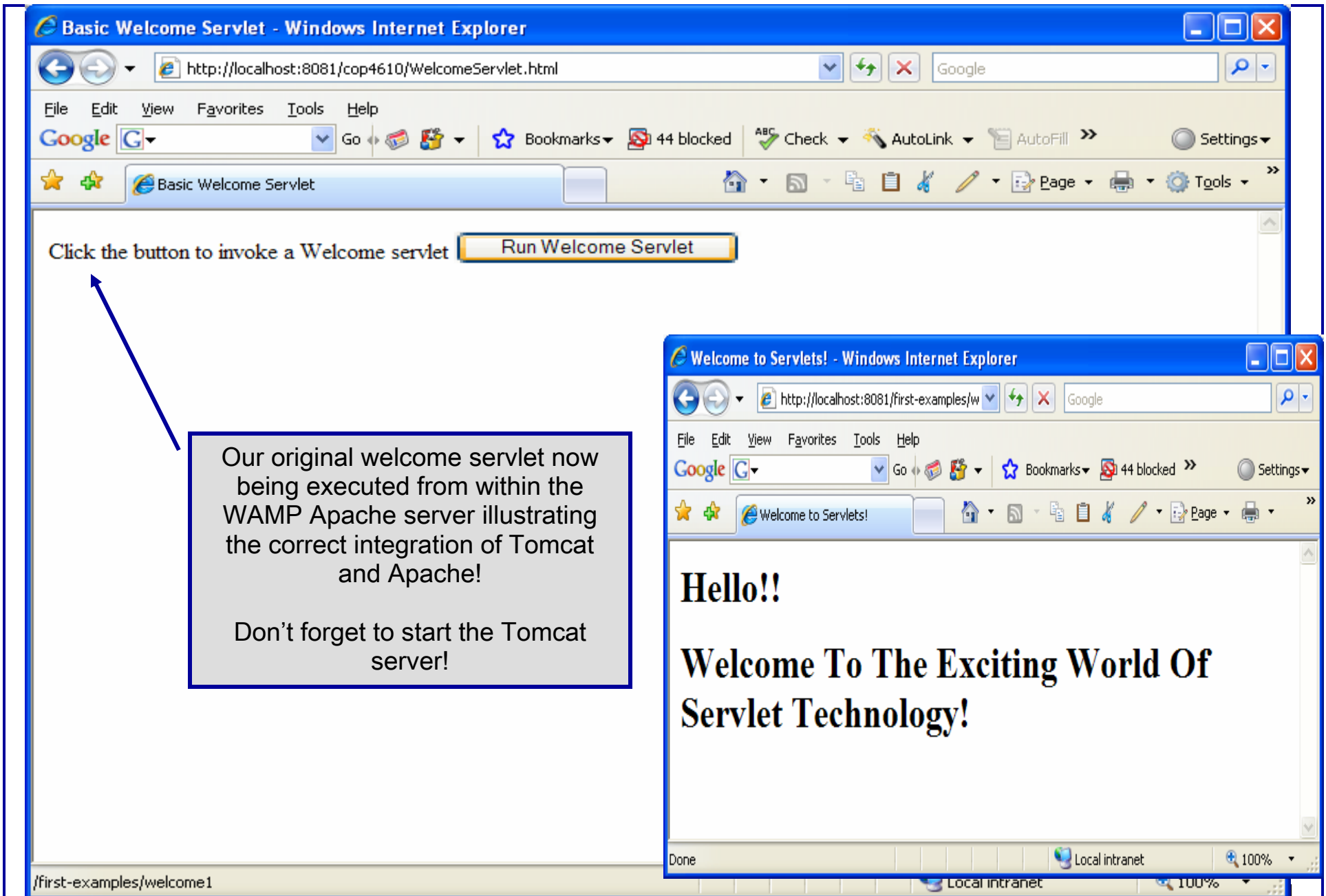
The portion of the Tomcat worker.properties file showing the definition for the default AJP 1.3 worker. NOTE: the port number listed here must match the port number listed in the Tomcat server.xml file as shown on the next page.



```
E:\Program Files\Apache Software Foundation\Tomcat 5.5.20\conf\server.xml - Windows Internet Explorer
E:\Program Files\Apache Software Foundation\Tomcat 5.5.20\conf\server.xml
File Edit View Favorites Tools Help
Google G Go 44 blocked Check AutoLink AutoFill Settings
Index of /apache/tomcat... E:\Program Files\Apac...
To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here for options...
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
<!-- Define a Proxied HTTP/1.1 Connector on port 8082 -->
<!-- See proxy documentation for more information about using this. -->
<!-- <Connector port="8082"
      maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
      enableLookups="false" acceptCount="100" connectionTimeout="20000"
      proxyPort="80" disableUploadTimeout="true" />
-->
<!-- An Engine represents the entry point (within Catalina) that processes
      every request. The Engine implementation for Tomcat stand alone
      analyzes the HTTP headers included with the request, and passes them
      on to the appropriate Host (virtual host). -->
<!-- You should set jvmRoute to support load-balancing via AJP ie :
      <Engine name="Standalone" defaultHost="localhost" jvmRoute="jvm1">
-->
<!-- Define the top level container in our container hierarchy -->
- <Engine name="Catalina" defaultHost="localhost">
  <!-- The request dumper valve dumps useful debugging information about
      the request headers and cookies that were received, and the response
      headers and cookies that were sent, for all requests received by
      this instance of Tomcat. If you care only about requests to a
```

The portion of the Tomcat server.xml file showing the definition for the AJP 1.3 connector. This connector port number listed here must match the port number in the workers.properties file as shown on the previous page.

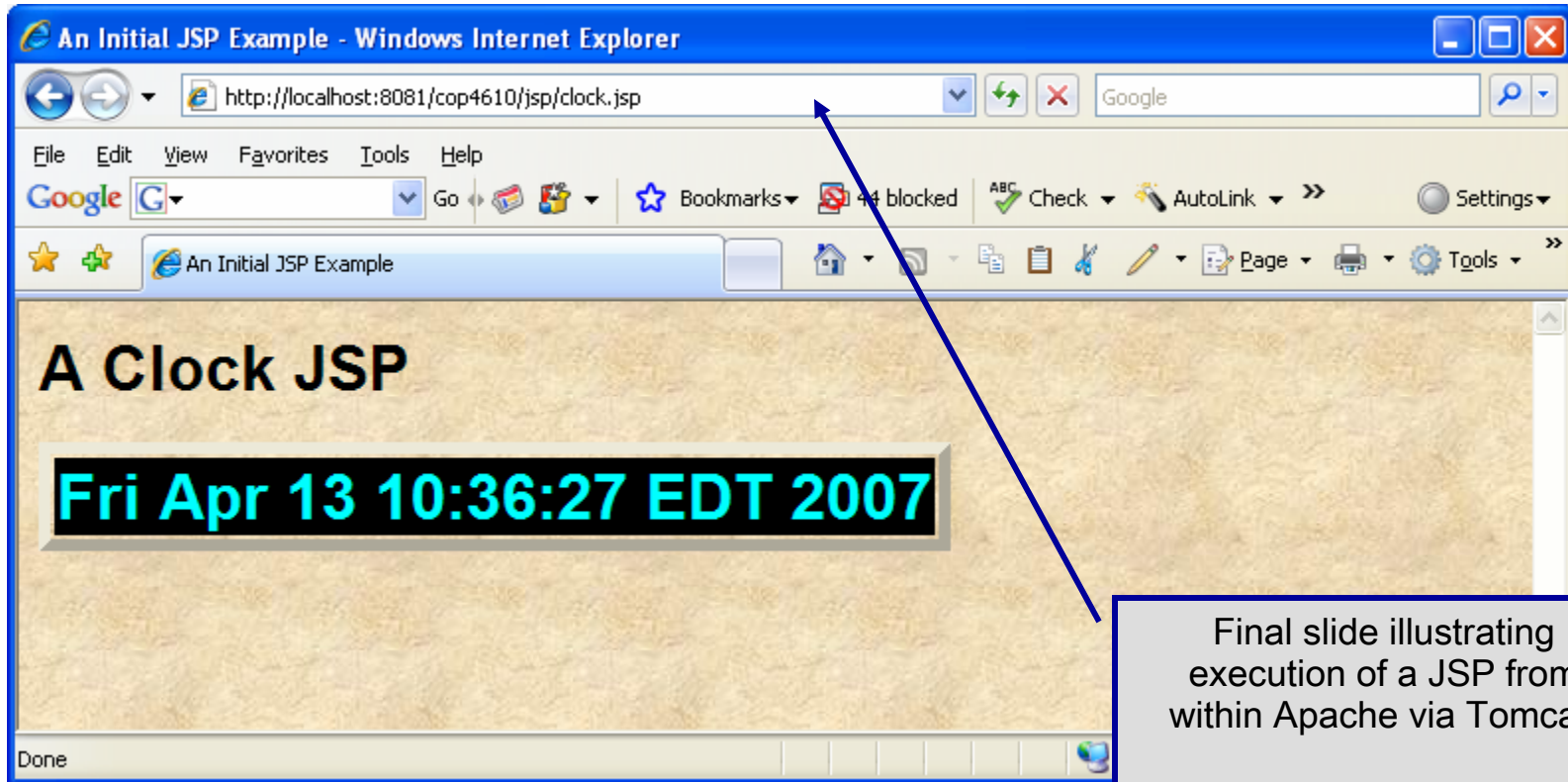




Our original welcome servlet now being executed from within the WAMP Apache server illustrating the correct integration of Tomcat and Apache!

Don't forget to start the Tomcat server!





Final slide illustrating execution of a JSP from within Apache via Tomcat.

