Installing and Using the xamppLite Standalone Web Server

You can install the required software quickly and easily. The version of XAMPPLite that is provided already includes the textbook sample files and code exercises that you will need. Refer to Appendix C if you:

- Want to learn more about your Web server.
- Want to install the latest version of the Web server
- Want to install the Web server on a Macintosh or Linux computer.
- Wish to download and install a **different** version of **xampp.**
- Intend to use a Web server that is **already installed** on your computer.
- Intend to use a Web server that is available for your use through an **Internet Service Provider** (ISP).

Installing XAMPPLite in Windows

It is a simple matter to install the XAMPPLite software. You can install the required files on a fixed drive on your personal computer or on a portable drive. Allow a minimum of 200MB disk space for the installation and for your work files. If you install XAMPPLite on a portable drive, you can then attach the drive to any computer in order to run the server and develop your code.

To install the server, simply create a work folder on the appropriate disk, then copy the file **webServer.zip** from your CD to the folder that you have created.

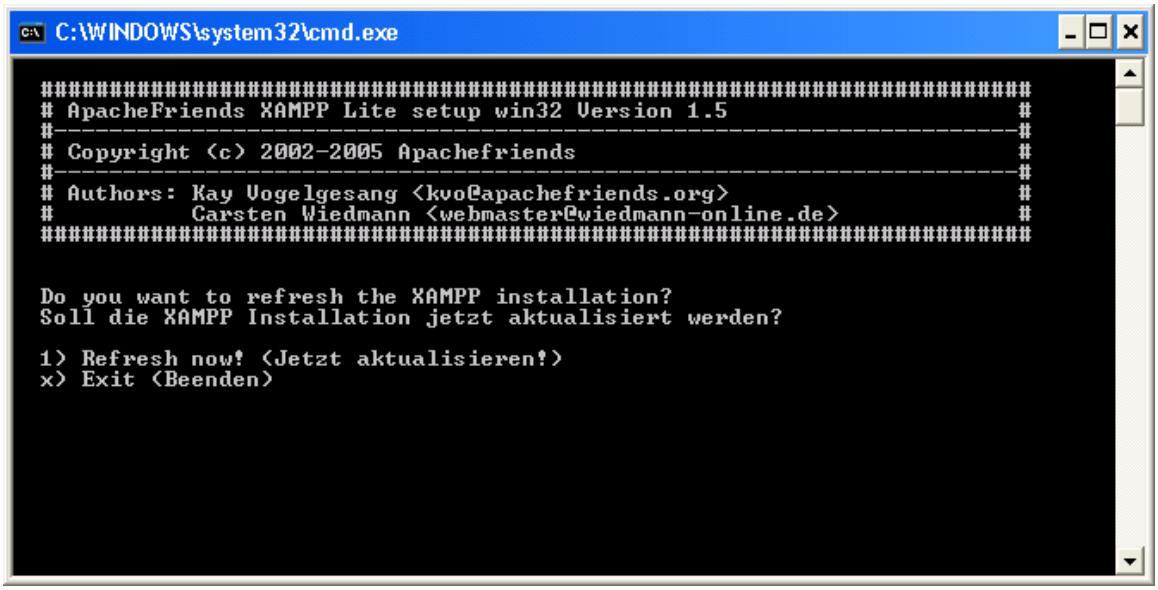
Now use Windows Explorer to navigate to your work folder and unzip the content of **xampplite.zip** to this folder. The installation process will create a new folder in your work folder named **xampplite**, which contains the entire **XAMPPLite** installation. That's it! The extraction process usually takes just a few minutes but may take considerably longer on some USB drives (that's because some of these drives are very inefficient when handling large numbers of small files).

The server installation includes the files and folders associated with the samples and chapter exercises in this book. These files are located in a folder named WebTech. If you navigate to the **xampplite\htdocs** folder you will find the WebTech folder, and inside this, you will find the **samples** and **coursework** folders that you will use. The WebTech folder must be located under the htdocs folder, otherwise the Web server will not be able to find them.

Running your Web server on Windows

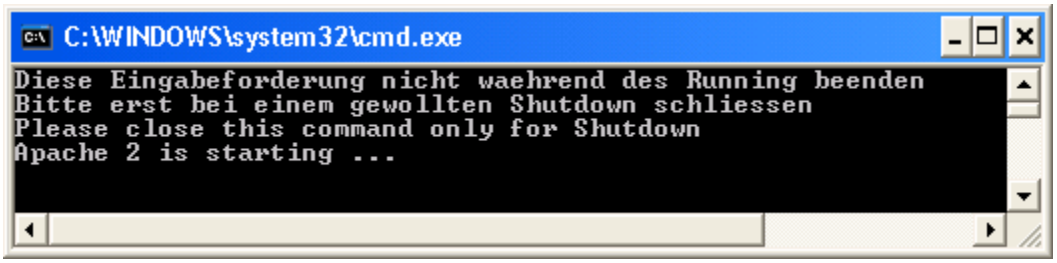
You are now ready to test your Web server. Navigate to the **xampplite** folder that was created when you unzipped your server. This folder contains a number of executable programs to manage your server. The first time that you run the Web server after installing XAMPPLite, you **must** first run the **setup** utility. To do this

simply double-click the file named **setup_xampp.bat**. When you do this, a DOS (console) window will open indicating that the setup was successful. You may be asked if you want to refresh the XAMPP installation. In this case, type **1** and press the **Enter** key.



You only need to run **setup_xampp.bat** the **first** time that you use XAMPPLite on any computer. However, if you run XAMPPLite on **multiple** computers from the **same** portable disk, you may get an error message when you attempt to start the server on a different computer that the one you normally use. That's because Windows may have assigned a different drive letter to your portable drive. If you get an error message, when you attempt to start your server, simply run **setup_xampp.bat** again - this will ensure that your file paths are set correctly for the new machine.

Now that your Web server has been configured for use you can start the Web server. To do this double click the **apache_start.bat** file in the **xampplite** folder. This will run the server and brings up a DOS window. Try this now.



That's it! Your Web server is now running. You may **minimize** this window, but do not **close** it while your Web server is running! The Web server must be running as long as you are working with your Web applications. When you are ready to shut down the Web server, double click the **apache_stop.bat** file in the **xampplite** folder. This will stop your Web server and close your DOS window.

IMPORTANT: always stop the Web server when you are finished working it, by double-clicking **apache_stop.bat**. Go ahead and stop your Web server now.

For help with installation problems, refer to Appendix C.

OK, let's test that your Web server is working as expected. Start your Web server, and start your Web browser. In your browser's address window type the following URL: **http://localhost/xampp**. This should open the XAMPP for Windows page.

If you see this page, your Web server is running successfully. Now that your Web server is running, you can test that the textbook **samples** folder is available for your use. Type the following URL in the address box:

http://localhost/Webtech/samples/welcome.html

This should bring up a welcome page with an interactive form.

Try completing the form and the press the "Submit the Form" button. If a second welcome screen appears in response to your submission, then everything is installed correctly and working fine.

NOTE: if you want to start and stop the Web server directly from your Desktop or Taskbar, right-click the **apache_start.bat** file and drag this to the Desktop or Taskbar, then release the mouse and choose **"Create Short Cut here"** from the menu that appears. Now do the same with the **apache_stop.bat** file.

Using URL's with your Web server

Every file on the Web has a unique URL, or Web address. The **URL** consists of a domain name, followed by the folders and a file name that indicate the location of the particular file on the server. Our local Web server has the domain name **localhost** and this domain name points to the folder **xampplite\htdocs** (the **xampplite** folder is located wherever you installed this software).

Earlier you typed the URL **http://localhost/WebTech/samples/welcome.html**. In this case you were directing your Web server to deliver a file named **welcome.html** to your Web browser. Since the domain name **localhost** references the folder **xampplite\htdocs**, the Web server finds this file in the following disk location:

xampplite\htdocs\Webtech\samples\welcome.html

The **xampplite** folder is located wherever you installed your Web server software.

We will keep our work files in **two** folders under the **xampplite\htdocs\Webtech** folder:

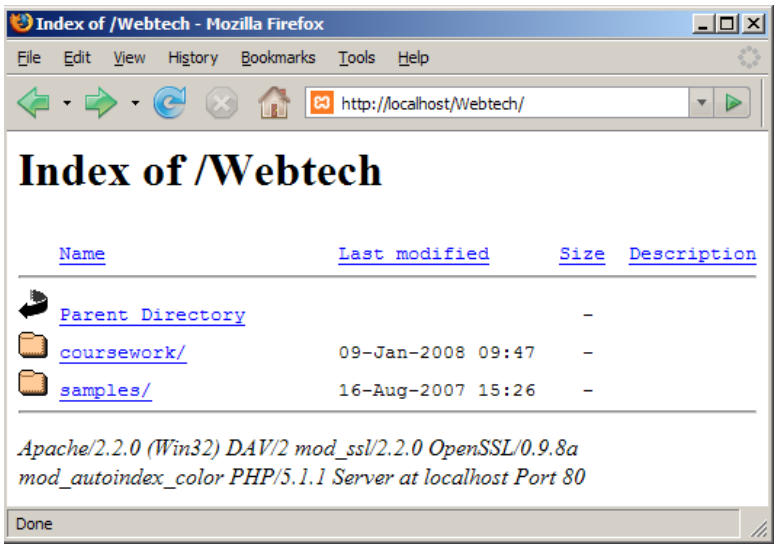
The **samples** folder will contain all of the sample files referenced in the textbook

The **coursework** folder will contain the code exercises that you complete as you work through each chapter of this book. This folder contains sub-folders for each chapter.

You have already seen how you can access a file in the **samples** folder. Similarly to access a file named **myFirst.php** in a folder named **Chapter02** inside the **coursework** folder, you would type the URL:

http://localhost/Webtech/coursework/Chapter02/myFirst.php

IMPORTANT! Always remember to first **start** your Web server **before** you attempt to run your programs. Always **stop** the Web server once you have completed your work. The URL to your programs will always begin with **http://localhost/** and this should be followed by the names of any subfolders, followed by the name of the file that you wish to open. Alternatively, you can click your way through folders to open the file you want. For example, if you just type **http://localhost/WebTech** the index of the Webtech folder will be displayed.



You can then click on the appropriate folder and locate the file that you wish to open.

The **samples** folder contains a large number of examples that we will use in this course. To see a list of all the files in the **samples** folder, type **http://localhost/Webtech/samples/** in your browser window. You will notice that many files are listed in pairs with the same name but two different extensions (**.html** and **.php**). In these cases, click the **.html** files rather than the **.php** files to see what they do (the **.html** files contain forms that are used to "drive" the PHP programs). Feel free to run as many programs as you like. You will explore these in detail as you work through this textbook.

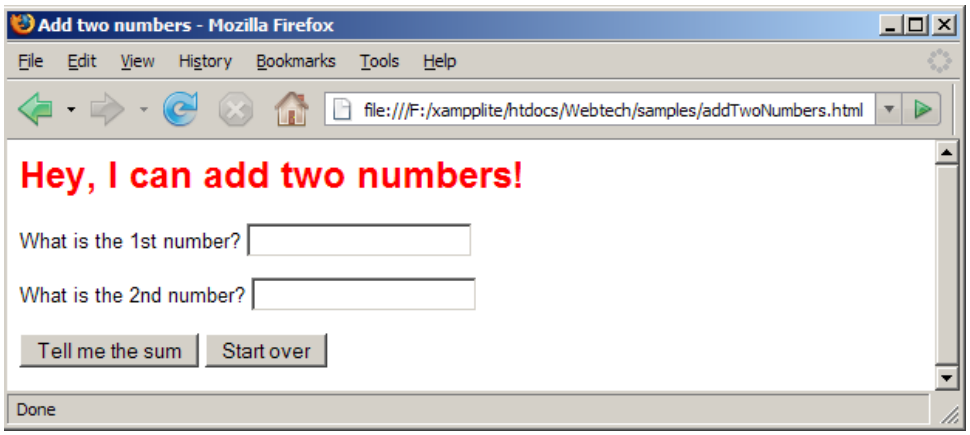
Always use URL's to run your Web applications!

Always run the Web server and use the **localhost** domain whenever you want to view files in your Web browser. Windows allows you to view HTML files using a Windows file path but that approach will not work for our purposes. To understand

this, let's try running a Web application using a Windows file path instead of a URL, just to see what happens.

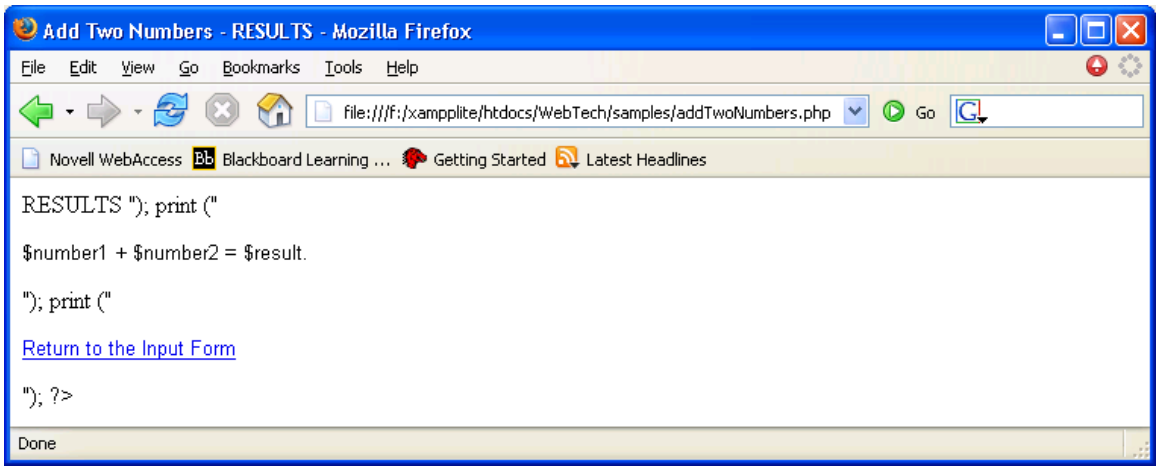
Navigate to the **samples** folder (xampplite\htdocs\Webtech\samples). Double-click the file named **addTwoNumbers.html**. A Web browser will probably start up, open the file and displays the Web page. This page looks fine, but notice the file path that is displayed in the browser's address window, which will be something like:

file:///F:/xampplite/htdocs/Webtech/samples/addTwoNumbers.html



Because we opened the file directly, we see the Windows file path in the address window rather than a URL. If your operating system associated a program other than a Web browser to handle HTML files then you may see something quite different.

Assuming your Web browser is displaying the page, if we now type in two numbers and click the "Tell the Sum" button, we will see something unexpected.



This doesn't make much sense! What is happening is that when the "Tell the Sum" button was clicked, the Web browser opened a file named **twoNumbers.php** which contains the PHP instructions to process the form. However the browser was simply opening the file on the disk and displaying the contents. The browser was **not** submitting a request to your Web server to open the file and execute the instructions. The Web server includes the PHP processor which can execute PHP

instructions. To access the Web server from your Web browser you **must** use URL's that begin with the domain name **localhost**.

The **correct** way to view your **.html** and **.php** files is to first run the Web server (if it is not already running), and then provide the URL to open the appropriate file. For example to correctly run the **addTwoNumbers** application, use the URL:

http://localhost/WebTech/samples/addTwoNumbers.html

To summarize: if you run a Web application and the screen shows you the PHP source code instead of the results of executing the PHP instructions, it means that you accessed the PHP file using its Windows file path instead of using a URL that includes the **localhost** domain name.

Where to save your files

You will create your own files in chapter folders under the **coursework** folder. You will use a text editor to develop your HTML and PHP code and save these files to locations under the **xampplite\htdocs\WebTech\coursework** folder. Once you are ready to test your code, be sure that the Web server is running and then use your Web browser to open your files. The URL for your files will be:

http://localhost/WebTech/coursework/ChapterXX/yyy

where **ChapterXX** is a chapter number (for example **Chapter02**) and **yyy** is the name of a specific file, for example **myFirst.html**. Alternatively you can just type **http://localhost/WebTech/coursework** and then click on the appropriate folder and file.

Creating an HTML document

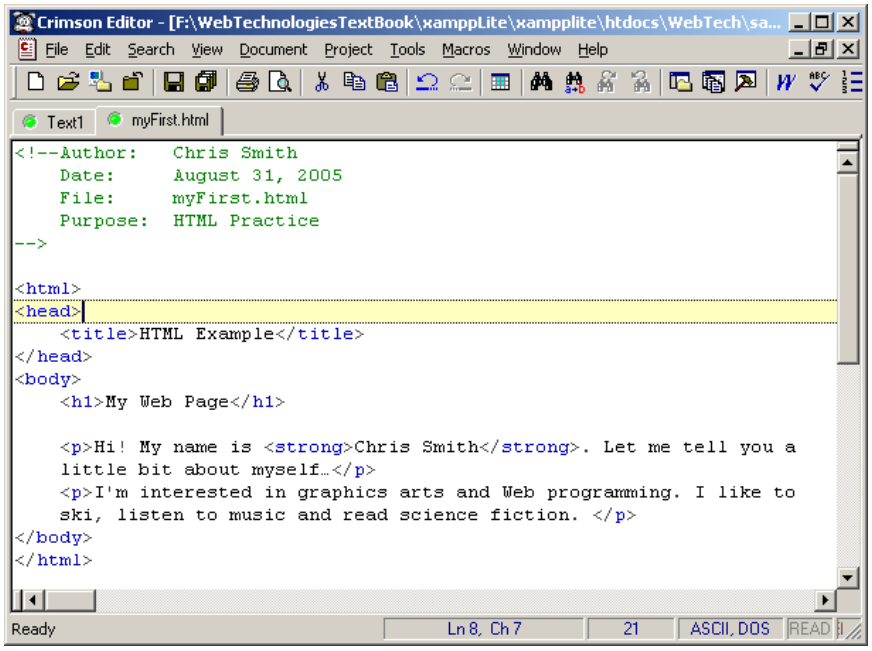
In order to get started, you will first create a simple HTML document, store it on your server and then send a request to open the document from your client Web browser. Don't be concerned about understanding this document right now – you will learn about HTML in Chapter 4.

Open **Crimson Editor** or any text editor that you wish to use. Type in the text for **myFirst.html**, but write your name instead of "YOUR NAME", write today's date instead of "TODAY'S DATE", and write something about yourself to replace the words "WRITE ABOUT YOURSELF HERE":

```
<!--   Author:   YOUR NAME
       Date:    TODAY'S DATE
       File:     myFirst.html
       Purpose: HTML Practice
-->

<html>
<head>
       <title>HTML Example</title>
```

```
</head>
<body>

	<h1>My Web Page</h1>

	<p>Hi! My name is <strong>YOUR NAME</strong>. Let me tell you a little
	bit about myself…</p>

	<p>WRITE ABOUT YOURSELF HERE</p>

</body>
</html>
```

This is how your file might look if you are using Crimson Editor as your text editor.



Choose **Save As** from the **File** menu and save the file as follows (your **Save in** address will reflect the actual location of your **xampplite** folder):

Save in: **xampplite\htdocs\WebTech\coursework\Chapter02**

File name: **myFirst.html**

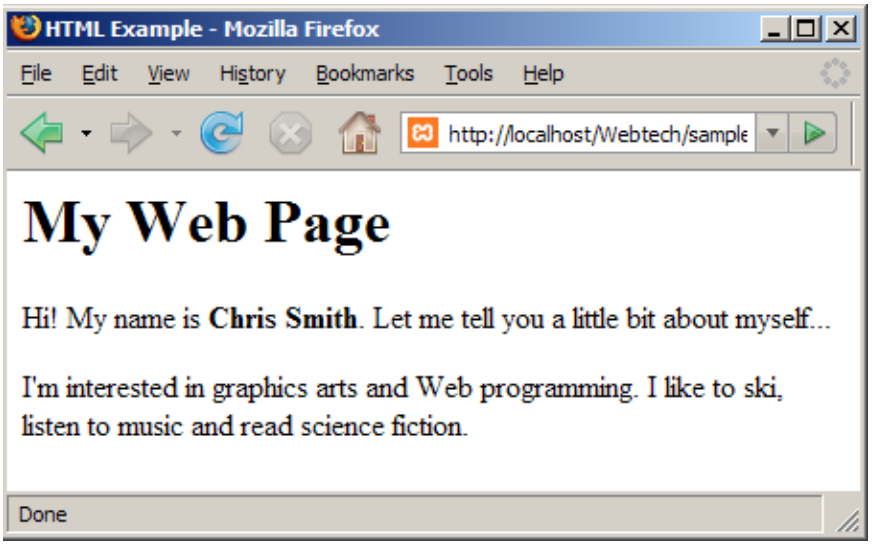
Save as Type: **HTML**

The file has now been stored on the Web server. You can now submit a request to view this file from your Web browser. Be sure your server is running. Now type the following URL in your browser's address box:

http://localhost/WebTech/coursework/Chapter02/myFirst.html

When the browser submits this request, the Web server receives the URL and locates the file (myFirst.html). Since the file has an .html extension, the server simply sends

the file contents back to the Web browser for display. Web browsers are designed to read HTML documents and treat any HTML tags as formatting instructions. We will learn more about this in the next chapter. Your HTML page should look similar to this screenshot:



If the link to **myFirst.html** does not work either your file name is different or the file is not in the correct location, or the Web server is not running.

If you wish to make changes to your document, simply edit your code in your text editor. Be sure to save your changes before viewing the file again and be sure to **refresh** the page in your Web browser otherwise your browser may continue to display the previous version.

Congratulations! You have just created a simple Web page, stored it on your local server, then accessed the page from a client (your Web browser)!

Creating a PHP program

Now let's create a simple PHP program. Don't be concerned about understanding this document right now – you will learn about PHP in Chapter 5. Type the code listing for **myFirst.php** in **Crimson Editor** (or your preferred text editor) exactly as written except type your name instead of "YOUR NAME" and today's date instead of "TODAY'S DATE". Note that you can copy and paste code from **myFirst.html** to save some time:

```
<!--    Author:   YOUR NAME
        Date:     TODAY'S DATE
        File:     myFirst.php
        Purpose: PHP Practice
-->

<html>
<head>
```

```
        <title>First PHP Example</title>
</head>
<body>
        <h1>Circle Calculation</h1>

        <?php

                $radius = 15.75;
                $area = pi() * pow ($radius, 2);
                $circumference = 2 * pi() * $radius;

                print("<p>A circle with a radius of $radius has an area of $area and
                a circumference of $circumference.</p>");

                print("<p>That's all that I have been designed to tell you!</p>");
        ?>

</body>
</html>
```

Choose **Save As** from the **File** menu and save the file as follows:

Save in: **xampplite\htdocs\WebTech\coursework\Chapter02**

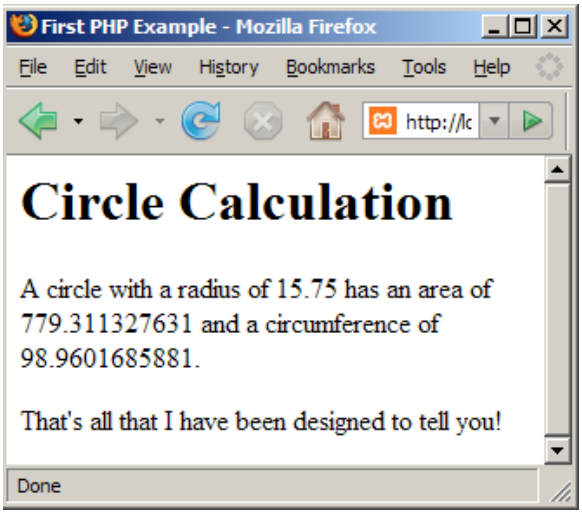
File name: **myFirst.php**

Save as Type: **PHP**

Now view this in your browser by typing the following URL:

http://localhost/WebTech/coursework/Chapter02/myFirst.php

When the browser submits this request, the Web server receives the URL and locates the file (myFirst.php). Since the file has an .php extension, the server runs a PHP processor to execute any PHP code in the file and assemble a new HTML document. Once the PHP has been completely processed, the newly created HTML document is sent back to the Web browser for display. Your page should look something like the following screenshot. We will learn more about this process in later chapters.



The text may wrap differently depending on the size of your browser window. If the link to **myFirst.php** does not work as expected, you may have used the wrong file name, or saved the files in the wrong location. Or you may have forgotten to start your Web server. You may receive an error message like this:

```
Parse error: parse error, unexpected T_VARIABLE in
F:\xampplite\htdocs\WebTech\coursework\myFirst.php on line 15
```

That means you have a syntax error in your PHP code. Programming languages such as PHP require a very precise syntax. There is a good chance that you may mistype something and your page may display differently (for example one or more numbers may not display as expected). Compare your code carefully with the example and see if you can find the errors. Once again remember to save your changes and remember to refresh the browser window to view your revised program.

Congratulations! You have just created a simple PHP program, stored it on your local Web server, and accessed it from a client (your Web browser)!

Creating an interactive HTML and PHP program

That last example displays information concerning a circle with a radius of **15.75**. We could improve the utility of this application by allowing the user to enter **any** radius. Next we will create a new version of this application that consists of two documents. The first (named **circle.html**) will be an HTML document that contains a form so that the user can submit a radius and submit this for processing. The second document (named **circle.php**) will contain a PHP program that receives the radius and calculates and displays the circumference and area of the circle. Here is the code for **circle.html**:

```
<!--    Author:   YOUR NAME
        Date:     TODAY'S DATE
        File:     circle.html
        Purpose: PHP Practice
```

```
-->

<html>
<head>
	<title>Circle Calculation</title>
</head>

<body>
	<h1>Circle Calculation</h1>

	<form action = "circle.php" method = "post" >
		<p>What is the radius of the circle?
		<input type = "text" size = "20" name = "radius" />
		</p>

		<p>
		<input type = "submit" value =
				"Tell me the area and circumference" />
		</p>
	</form>
</body>
</html>
```

Choose **Save As** from the **File** menu and save the file as follows (your **Save in** address will reflect the actual location of your **xampplite** folder:

Save in: **xampplite\htdocs\WebTech\coursework\Chapter02**

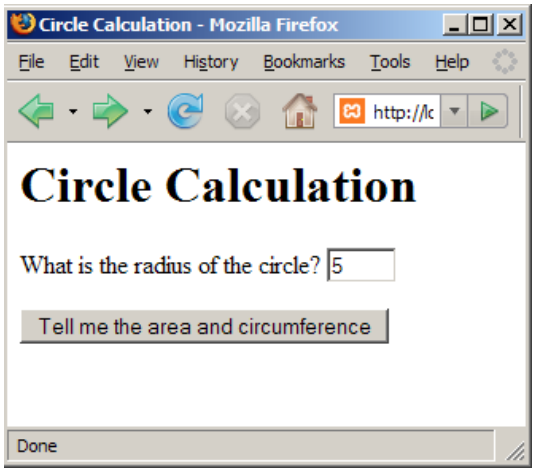
File name: **circle.html**

Save as Type: **HTML**

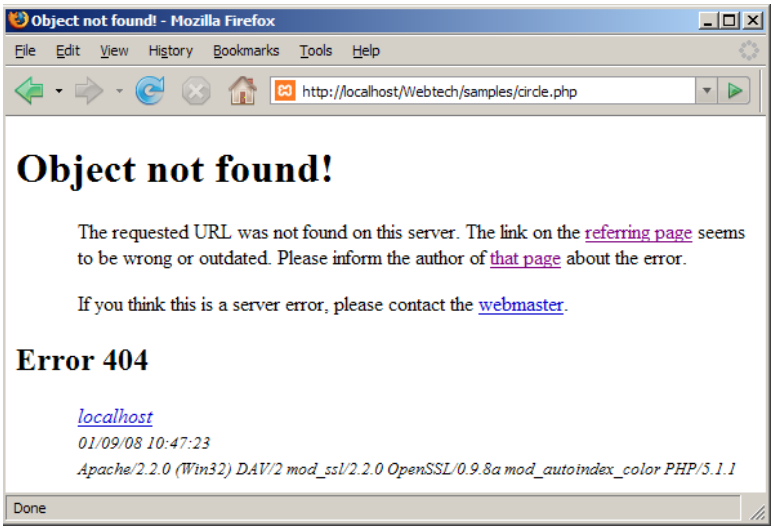
The file has now been stored on the Web server. You can now submit a request to view this file from your Web browser. Just type the following URL in your browser's address box:

http://localhost/WebTech/coursework/Chapter02/circle.html

If you do this you the document will display. Assuming that you typed everything correctly, you will that it contains will see a Web page with a form.



If you enter a radius in to the text box and click the "Tell me the area and circumference" button, you will get an error message, similar to that shown here:



That's because the form on this Web page is designed to send the radius to a program named **circle.php** in order for it to be processed. The problem is that we haven't created the **circle.php** program yet! So let's do that right now!

Here is the code for **circle.php**:

```
<!--    Author:   YOUR NAME
        Date:     TODAY'S DATE
        File:     circle.php
        Purpose: PHP Practice
-->

<html>
<head>
        <title> Circle Calculation</title>
</head>
<body>
```

```
        <h1>Circle Calculation</h1>

        <?php
                $radius = $_POST['radius']; // receive  the radius from the form

                $area = pi() * pow ($radius, 2);
                $circumference = 2 * pi() * $radius;

                print("<p>A circle with a radius of $radius has an area of $area and
                a circumference of $circumference.</p>");
        ?>

        <a href = "circle.html">Calculate another circle?</a>

</body>
</html>
```

Code Example: circle.php

Choose **Save As** from the **File** menu and save the file as follows:

Save in: **xampplite\htdocs\WebTech\coursework\Chapter02**

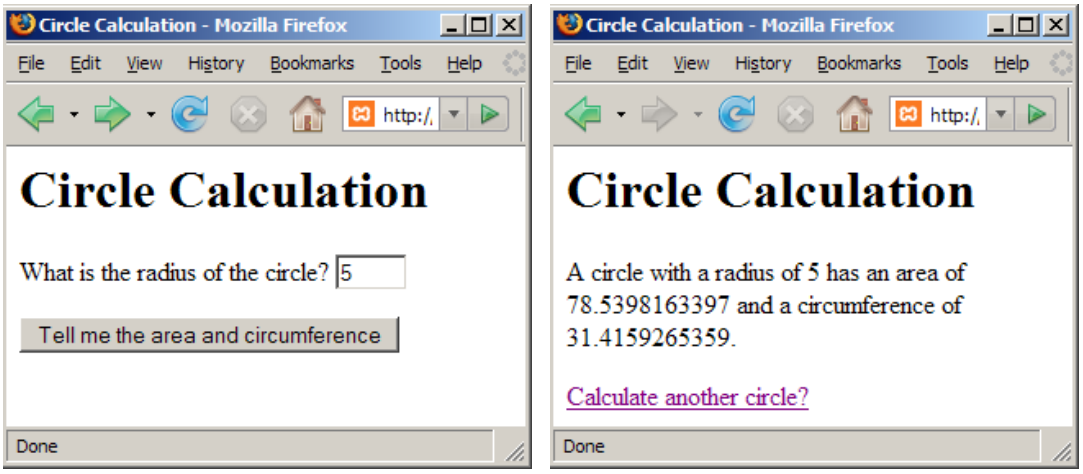
File name: **circle.php**

Save as Type: **PHP**

Now you should be able to use your form correctly. Open your Web document again:

http://localhost/WebTech/coursework/Chapter02/circle.html

Type a radius into the text box and click the "Tell me the area and circumference" button. This time you should see a new page that displays the area and circumference of a circle with the radius that you submitted.



Note that you can click the "Calculate another circle?" link to return to the first page.

Congratulations! You have now created a simple Web application that: provides an input form, processes the input, and displays the result.

Do not be concerned about how these documents actually work at this point. The purpose of the current exercise is to give you practice using a text editor to type HTML and PHP code, saving your documents to the correct location on your disk, making corrections as needed, and viewing your applications in a Web browser using the correct URL. In the following chapters you will learn how to design and create Web applications that include HTML pages with forms and PHP programs that process these forms.