Visibooks illustration-based computer textbooks

THE VISIBOOKS GUIDE TO PHP Basics

See. Do. Learn.

Table of Contents

Learning the Basics

In this section, you'll learn how to:

- **Install an FTP program**
- **Create a simple script**
- **Upload a script**
- **Run a script from a Web page**
- **Insert comments**
- **Format output with HTML**

Install an FTP program

1. Open your Web browser and go to:

   www.ipswitch.com

2. Download and install WS_FTP Home.

WS_FTP

FTP stands for File Transfer Protocol, a way to transfer files between computers over the Internet.

Using an FTP program is the most straightforward way to upload a Web site to a Web server. WS_FTP is the most popular FTP program used to upload and download Web pages.

The Home version is free to use for 30 days, and can be downloaded at www.ipswitch.com.

3. Open WS_FTP Home.

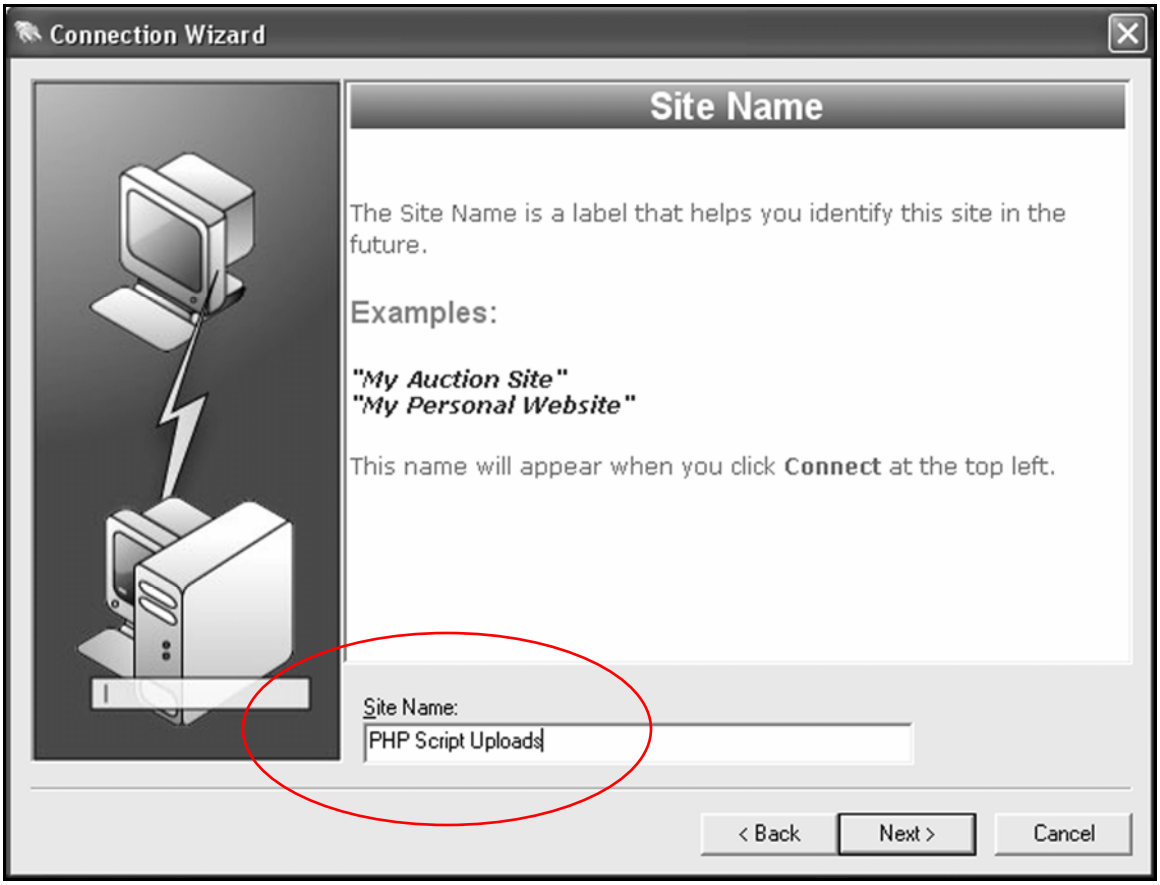
The Connection Wizard should open.

4. Click the Next > button.

5. When the **Site Name** screen appears, type:

PHP Script Uploads

in the **Site Name** box.



Then click the Next > button.

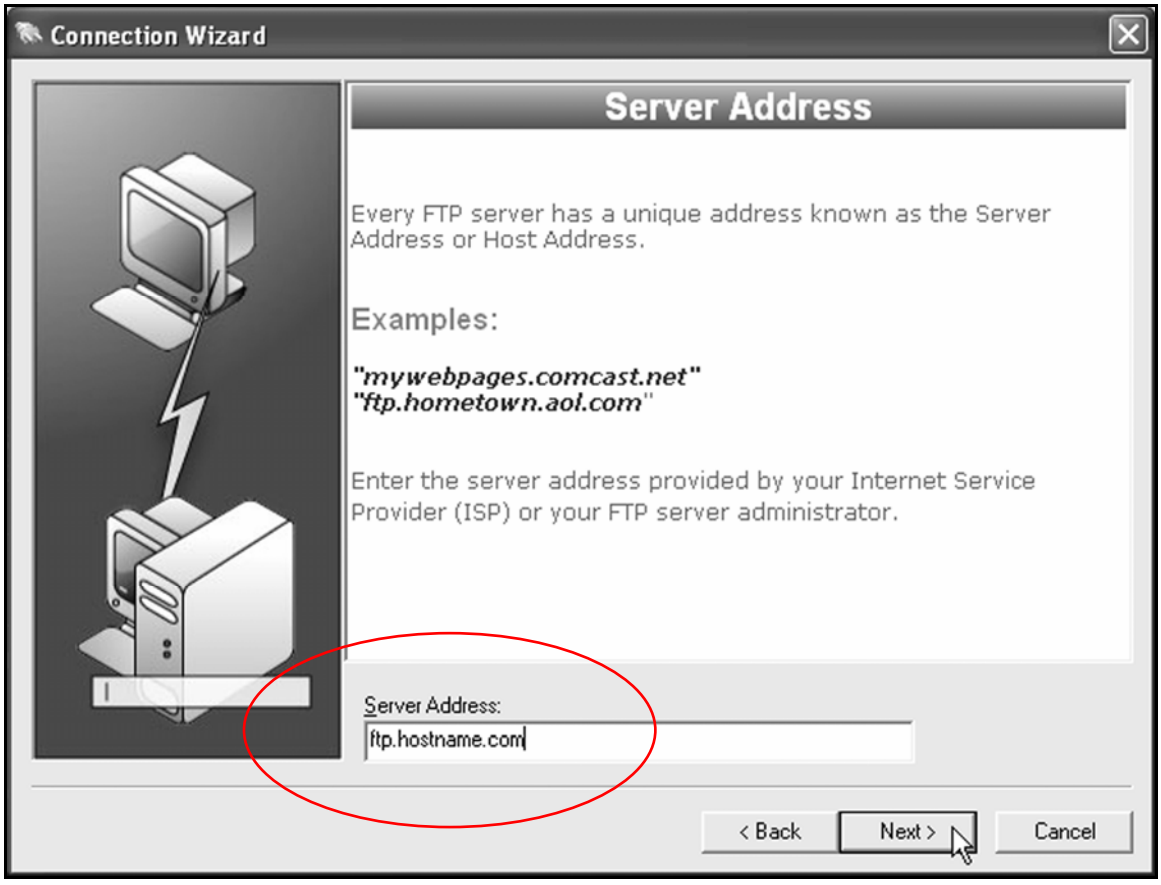
6. When the **Server Address** screen appears, type the host address of your server in the **Server Address** box.

It can be something like:

www.visibooks.com

washington.patriot.net

207.176.7.217



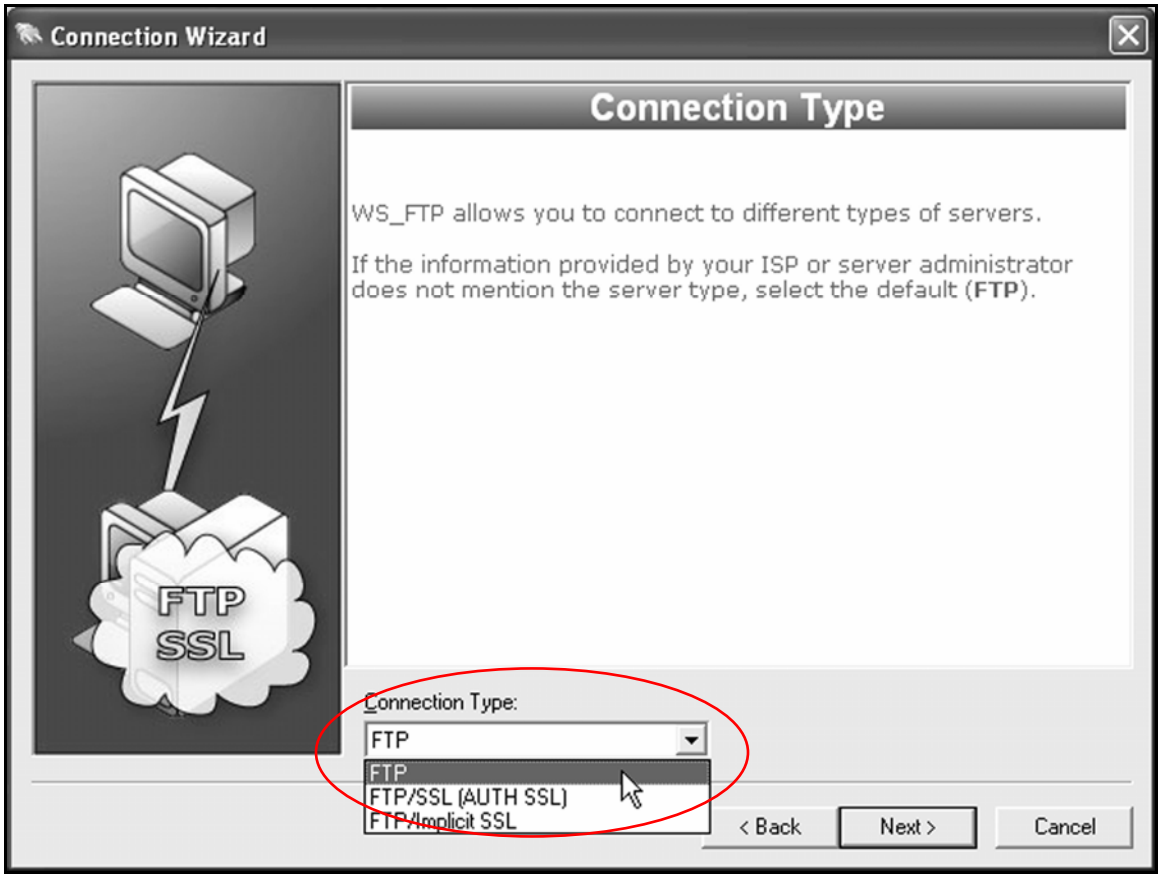
Then click the Next > button.

Tip: *You can get the Server Address of your Web site, as well as your username and password, from your Web server administrator.*

7. When the **User Name and Password** screen appears, type in your username and password.

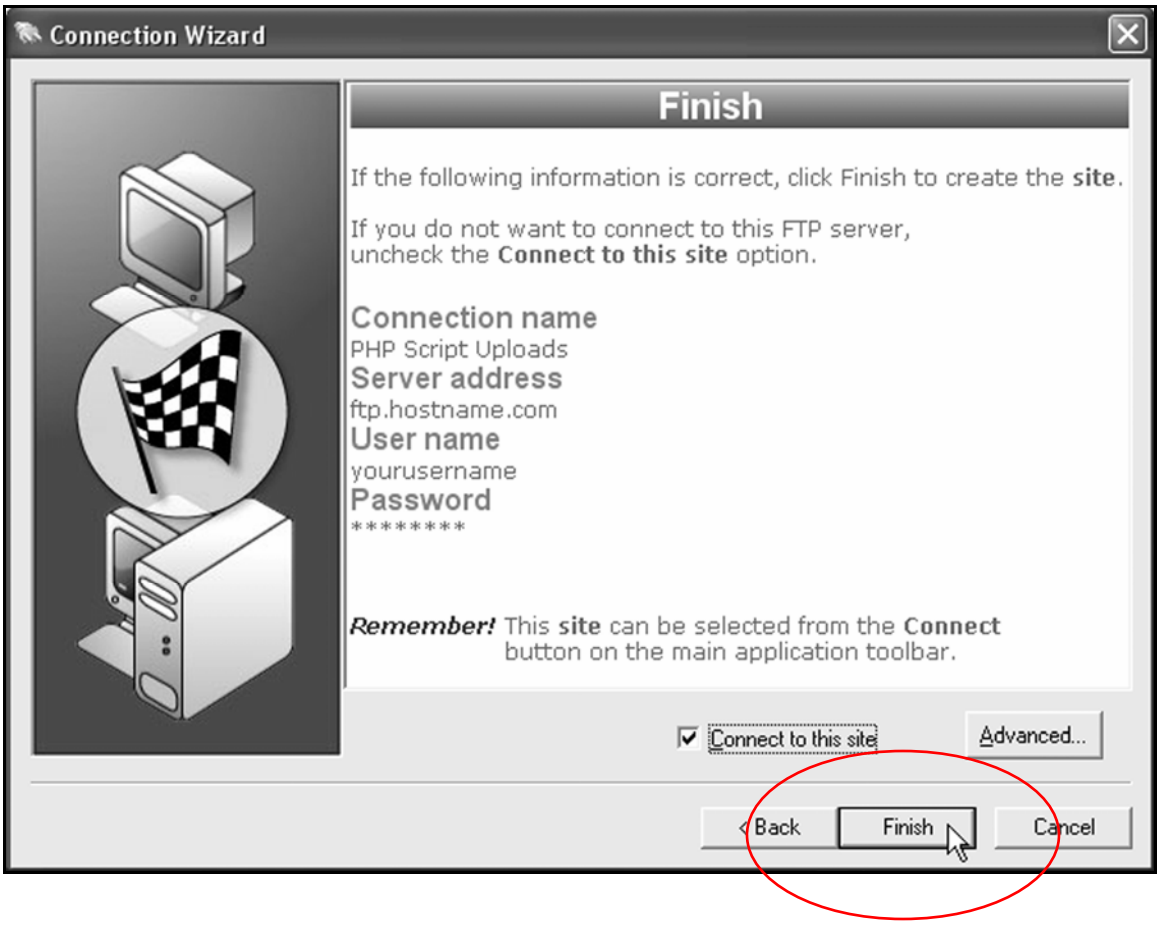
Then click the Next > button.

8. When the **Connection Type** screen appears, leave the connection type set at **FTP**.

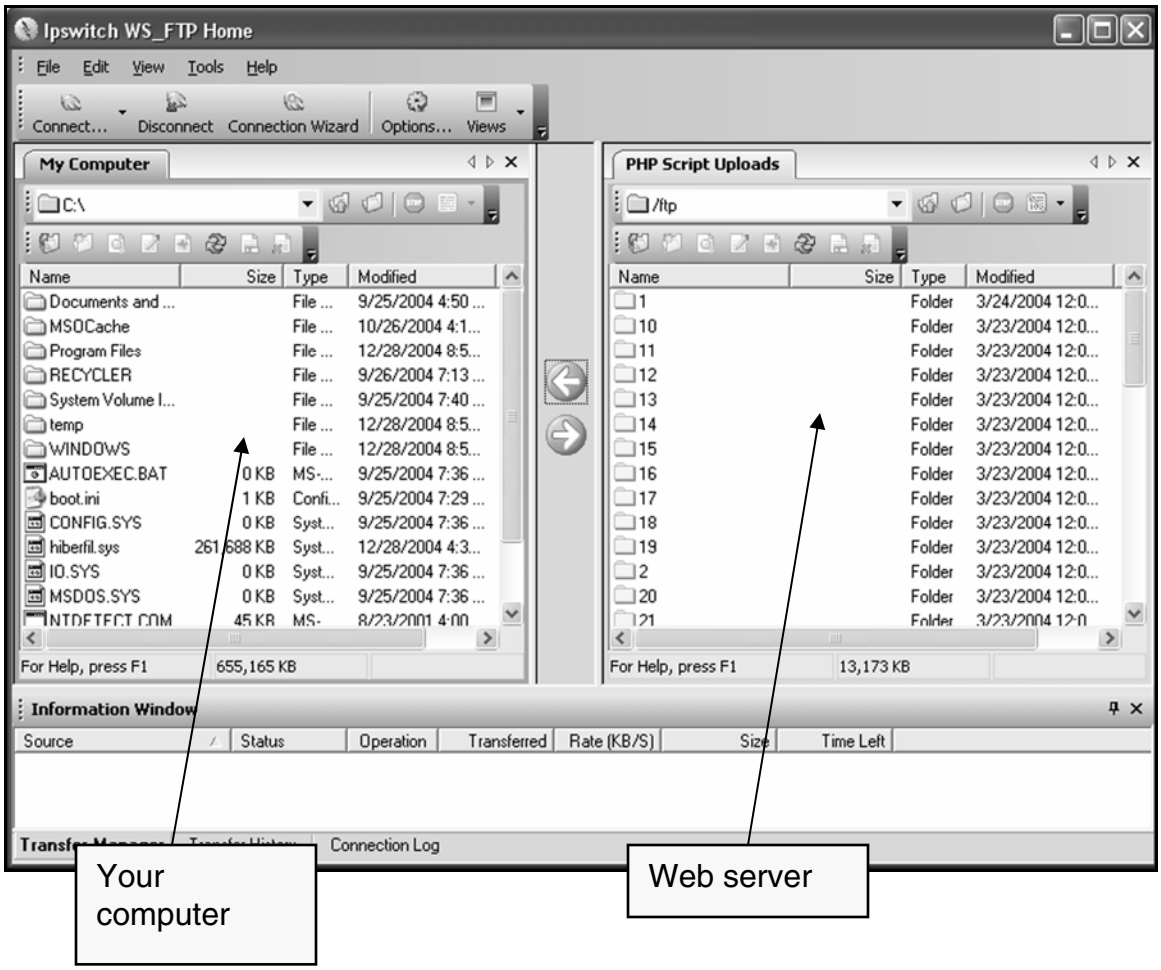


Then click the Next > button.

9. When the **Finish** screen appears, click the Finish button.



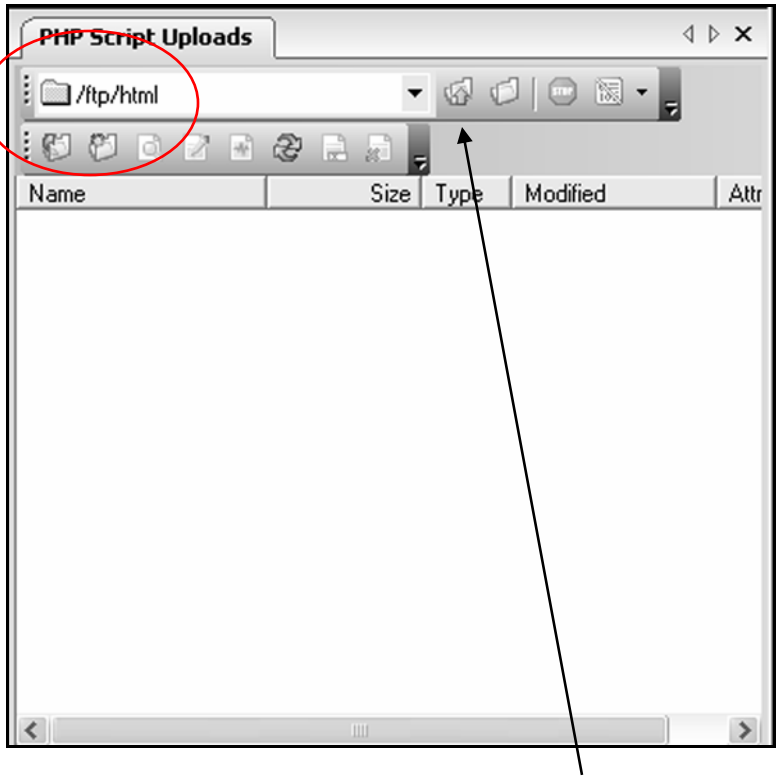
WS_FTP should connect to your Web server:



10. In the right-hand **PHP Script Uploads** pane, double-click on the **public_html** folder, **html** folder, or the folder that contains your Web pages on the server.

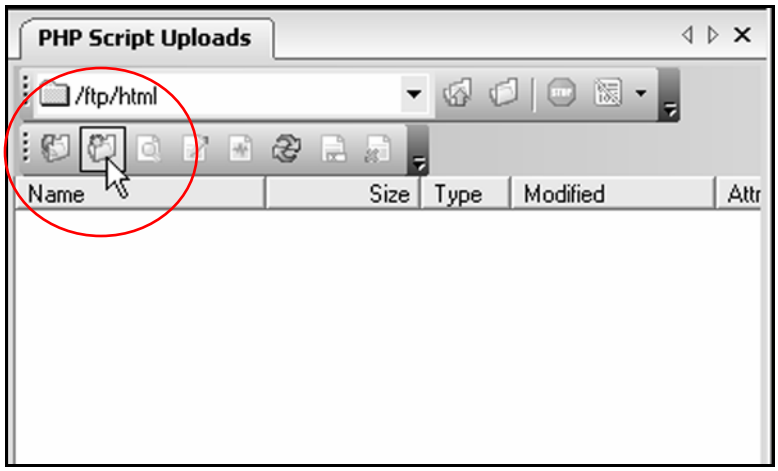
You should now see the contents of your Web site on the server.

11. In the right-hand **PHP Script Uploads** pane, navigate to that directory in your Web site.



Tip: *You may have to click the*  *icon to move up in the site hierarchy.*

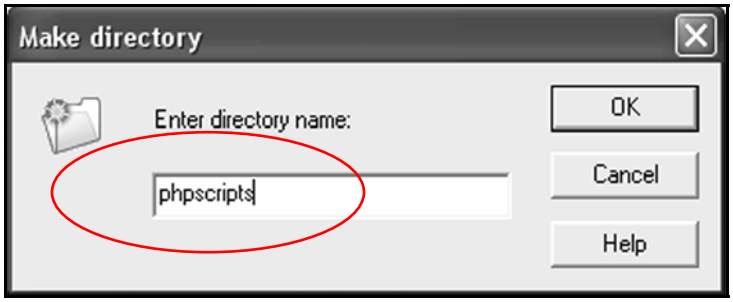
12. Click the  icon.



13. When the **Make directory** window appears, type:

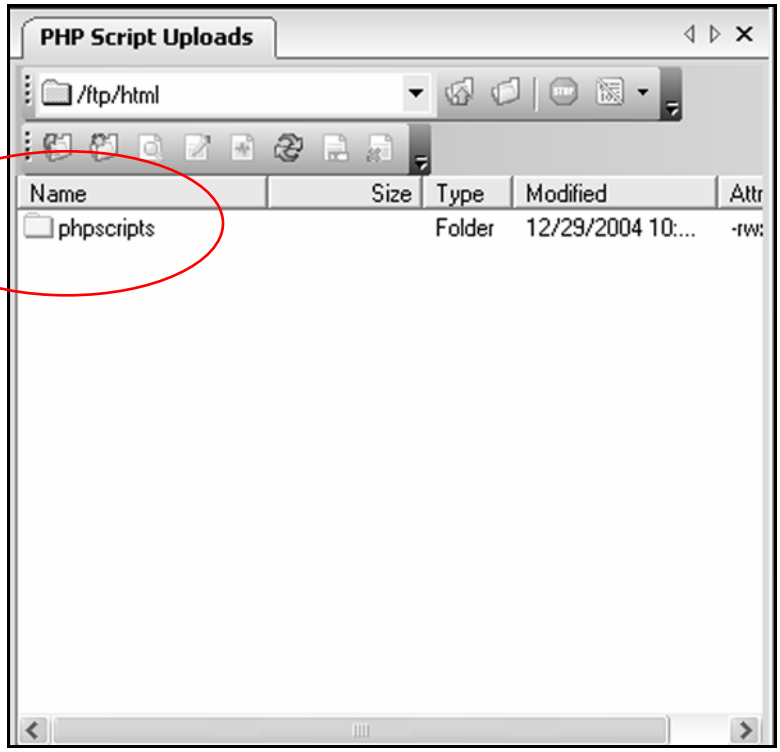
phpscripts

in the textbox.



14. Click the OK button.

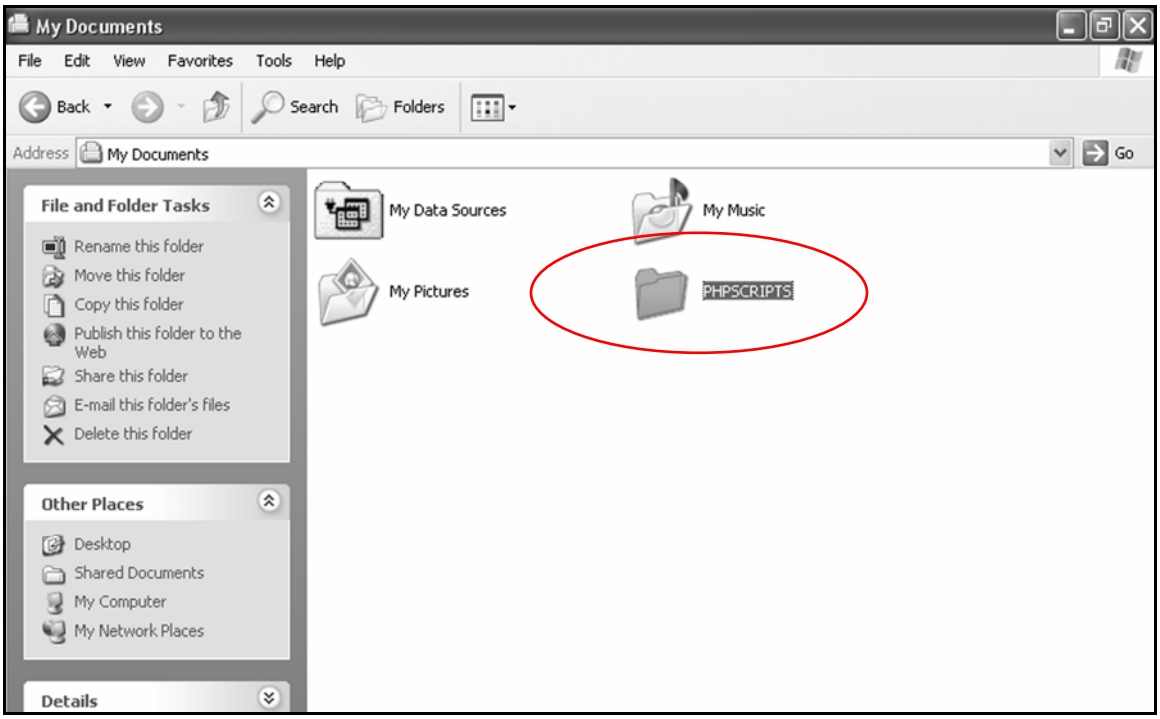
You should now see a directory called **phpscripts** in the right pane:



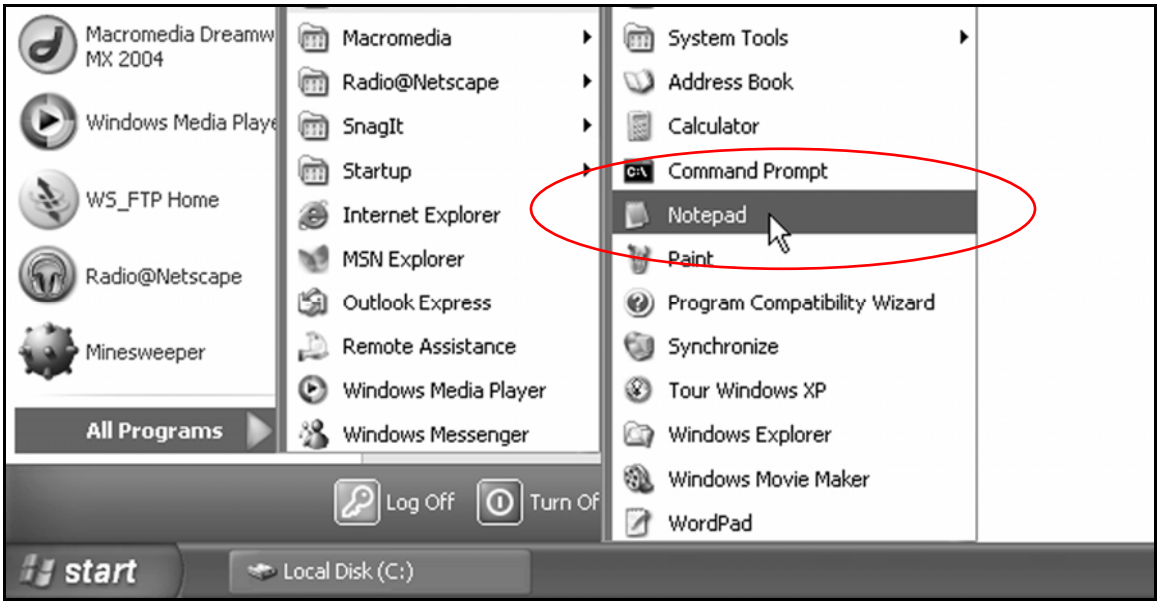
15. Close WS_FTP.

Create a simple script

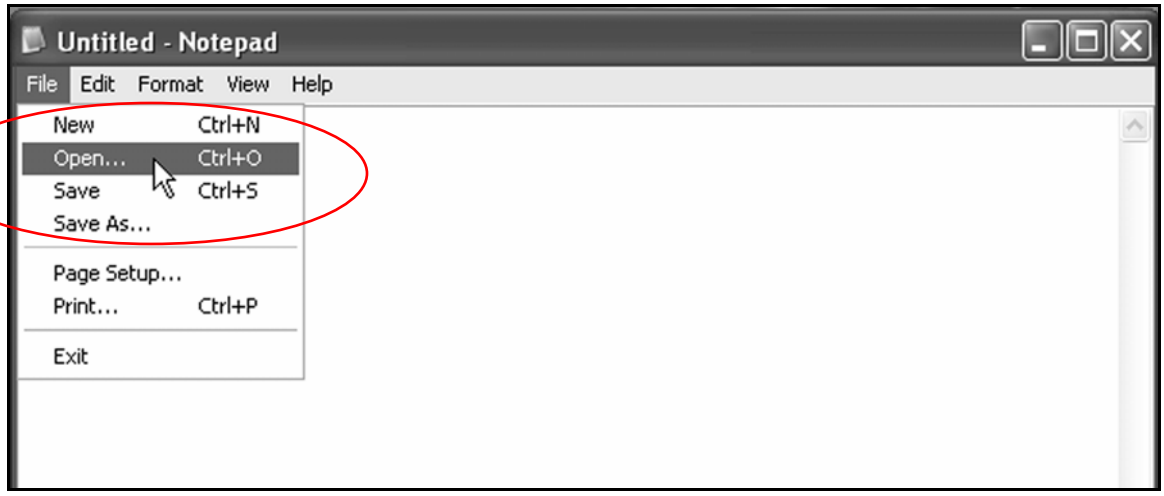
1. Create a folder called **PHPSCRIPTS** on your hard drive.



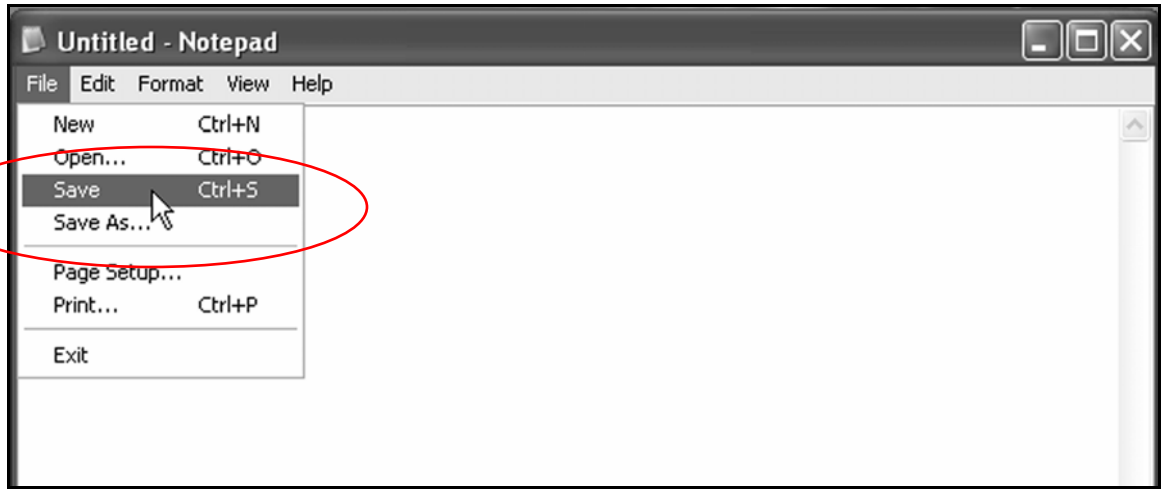
2. Open the Notepad program on your computer.



3. Click **File**, then **Open**.

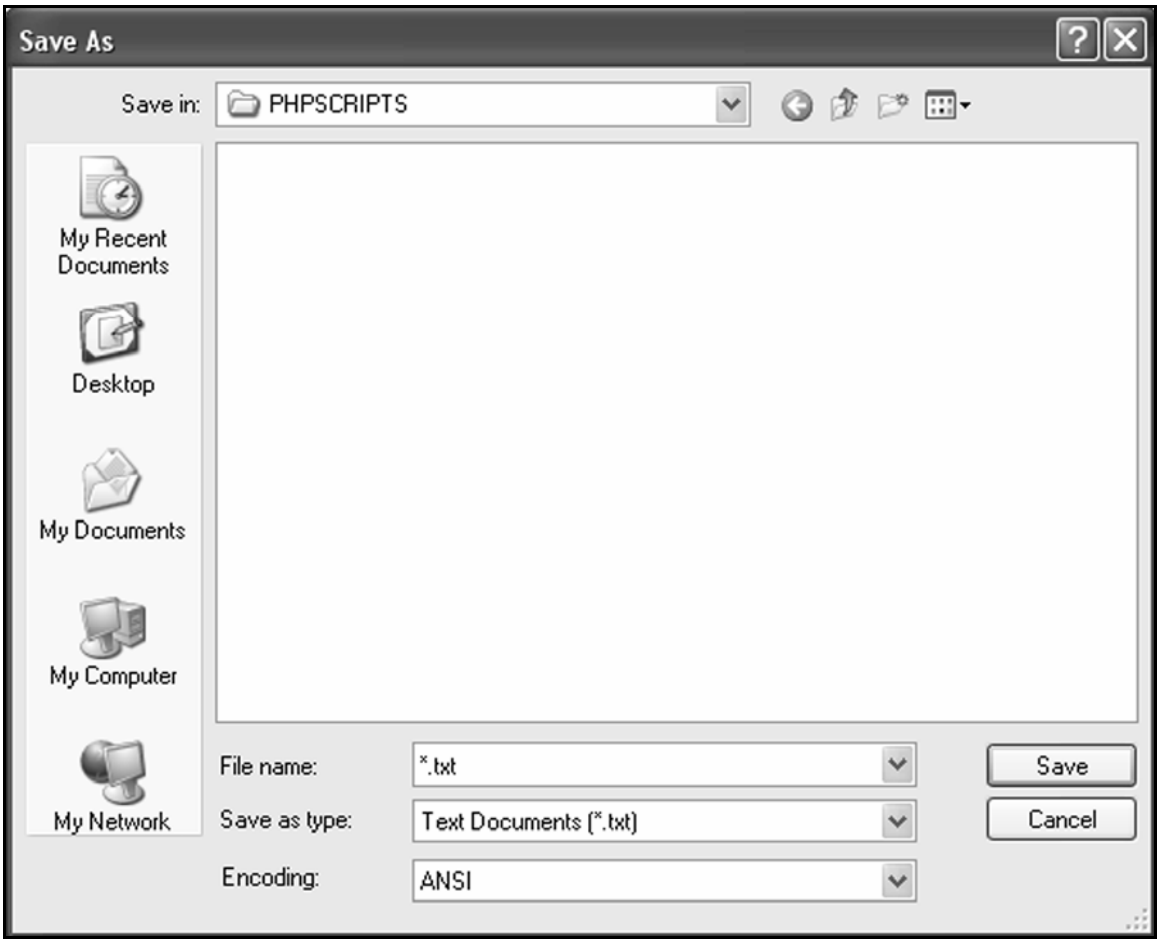


4. Click **File**, then **Save**.



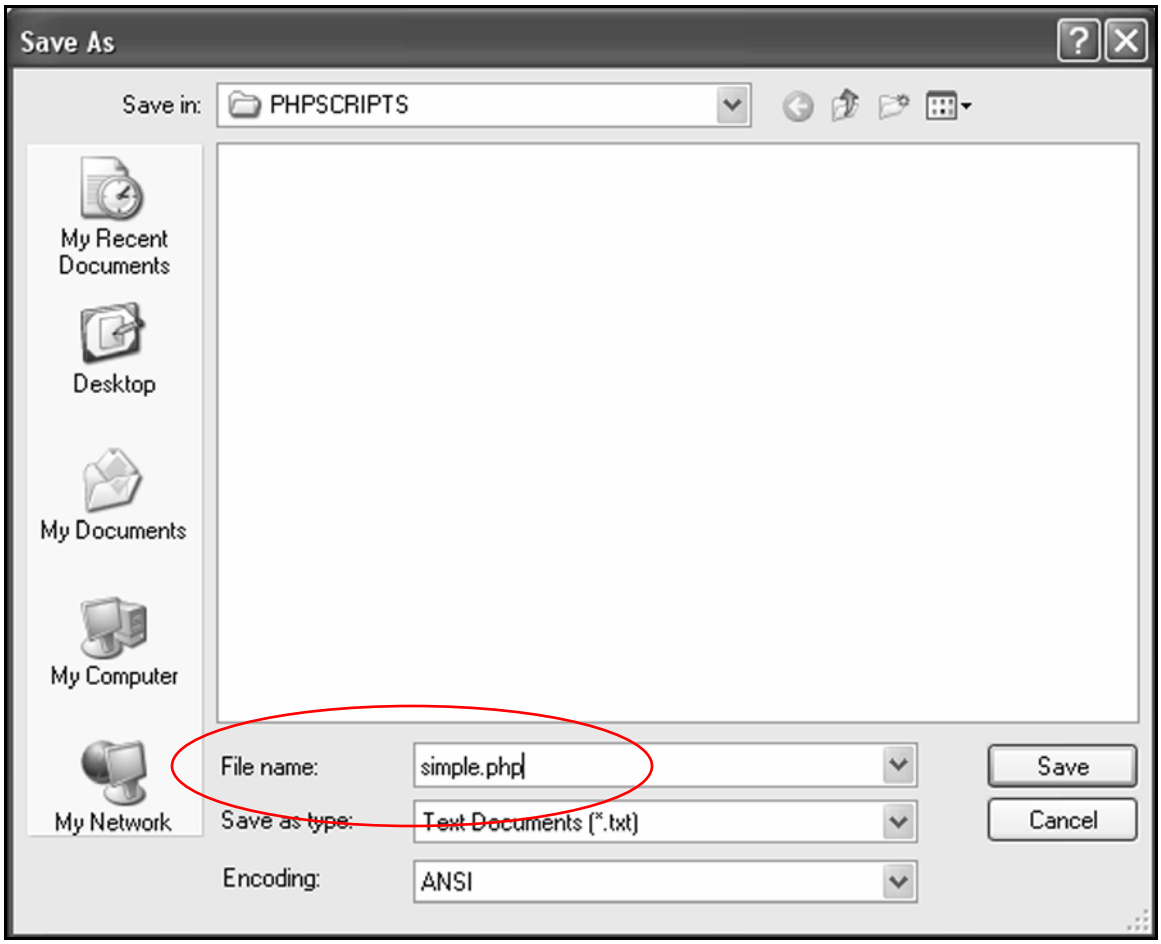
5. When the **Save As** window appears, navigate to the **PHPSCRIPTS** folder, then double click it.

The **PHPSCRIPTS** folder should appear in the **Save In** box.



6. In the **File Name** textbox, type:

simple.php



7. Click the Save button.

8. In the blank document window, type:

```
<?php

print "Welcome to ACME AUTO";

?>
```

Tip: *You're now typing commands to the Web server in the PHP language. Sometimes these commands are case-sensitive. Use lower-case for PHP commands—that is, everything not enclosed in quotation marks, like*

```
print "Welcome to ACME AUTO";
```

Also, don't forget to type a semicolon (**;**) at the end of each line. For your commands to work, or "execute," they need a semicolon (**;**) at the end.

Forgetting a semicolon is the most common programming mistake in most computer languages.

9. Save the script.

Here's what each line of this PHP script does:

- **`<?php`**

  This is the opening PHP tag. PHP code is always written between the opening and closing PHP tags.

- **`(blank line)`**

  Before the next line of code is a blank line. You can use blank lines throughout your PHP scripts.

  Blank lines allow you to group sections of code together, which makes scripts easier to read.

- **`print "Welcome to ACME AUTO";`**

  This print command tells the Web server to "print" the words between the quotes to the browser window.

  Remember: for a command string to execute, there must be a semicolon (`;`) at the end.

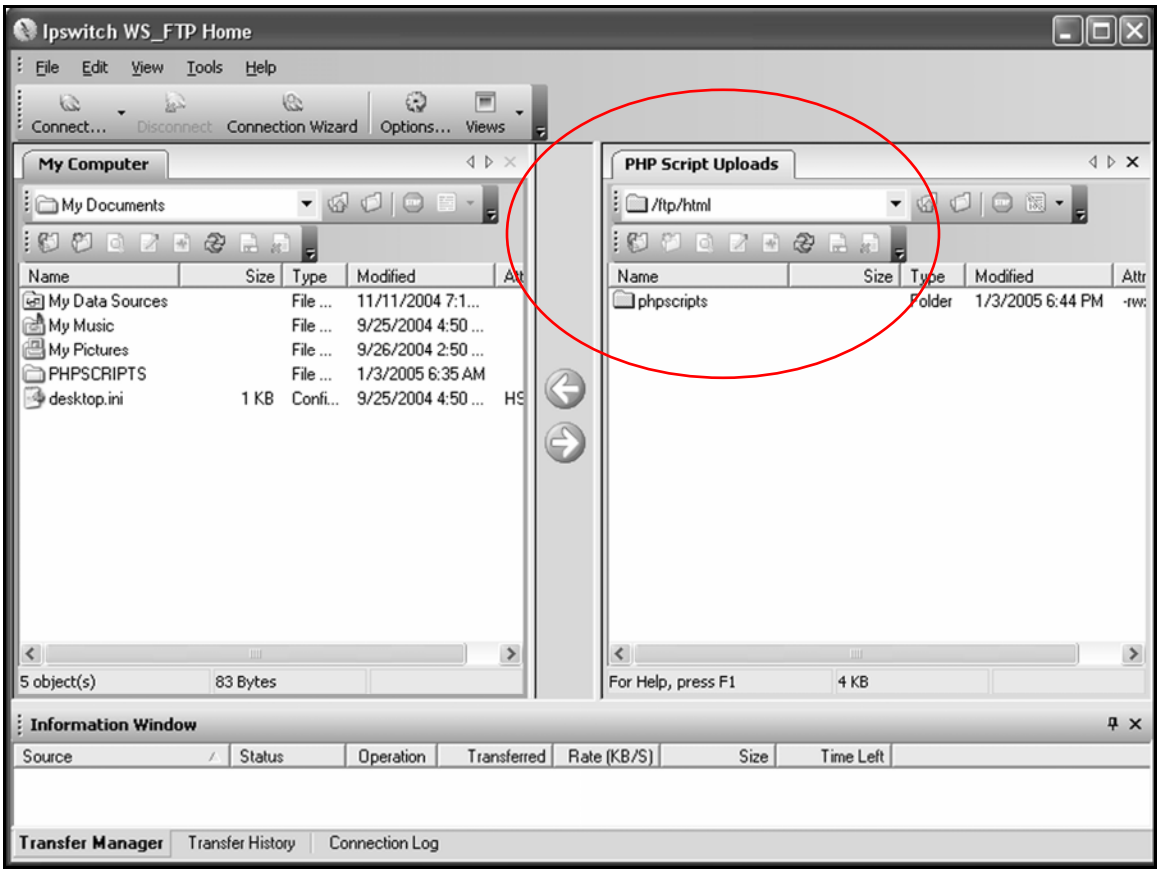
- **`?>`**

  This is the closing PHP tag. No more PHP code can be written after this closing tag without another opening PHP tag.

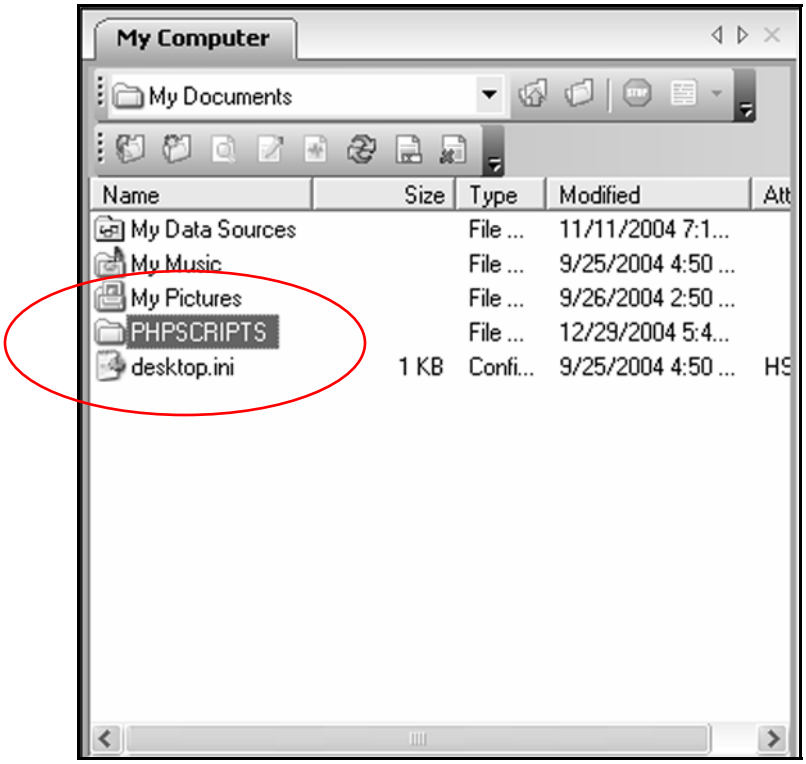
Upload a script

1. Open WS_FTP and navigate to the home directory on your Web server.

   It should look something like this:

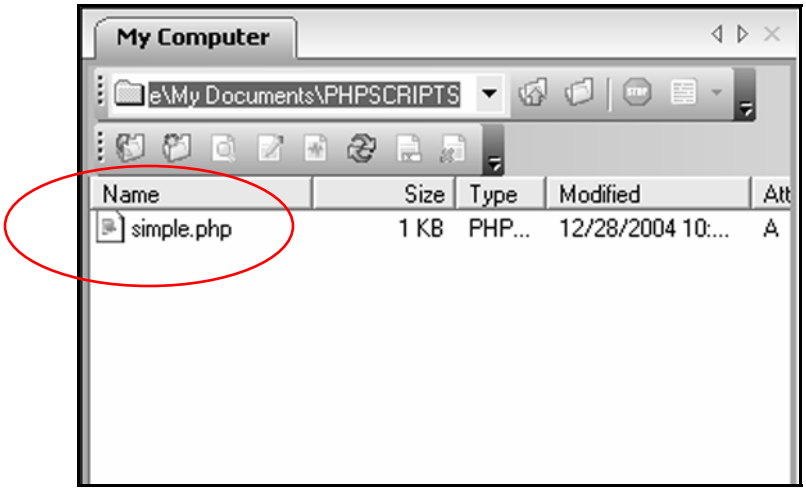


2. In the left-hand **My Computer** pane, navigate to the **PHPSCRIPTS** folder on your computer.

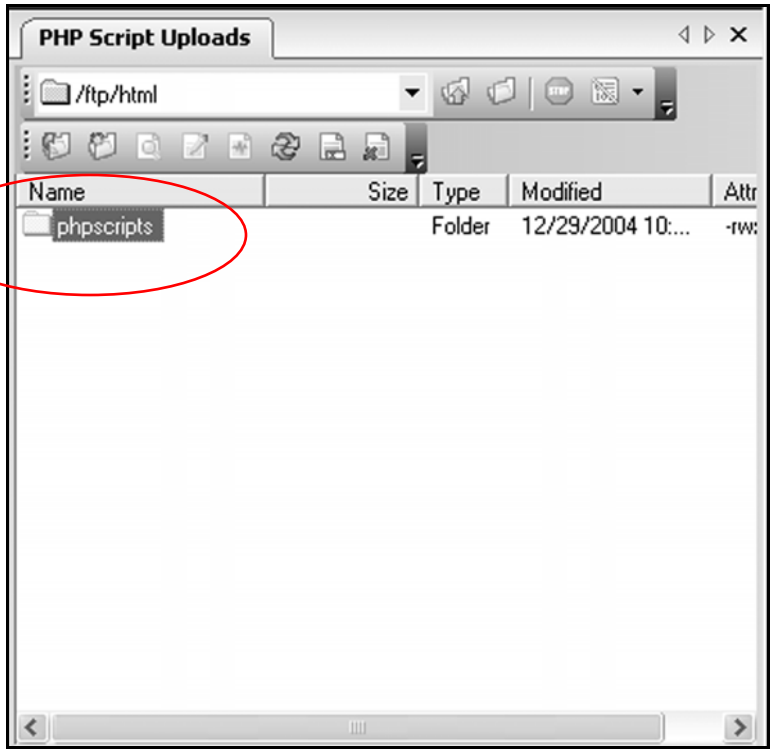


3. Double-click the **PHPSCRIPTS** folder.

simple.php should appear.

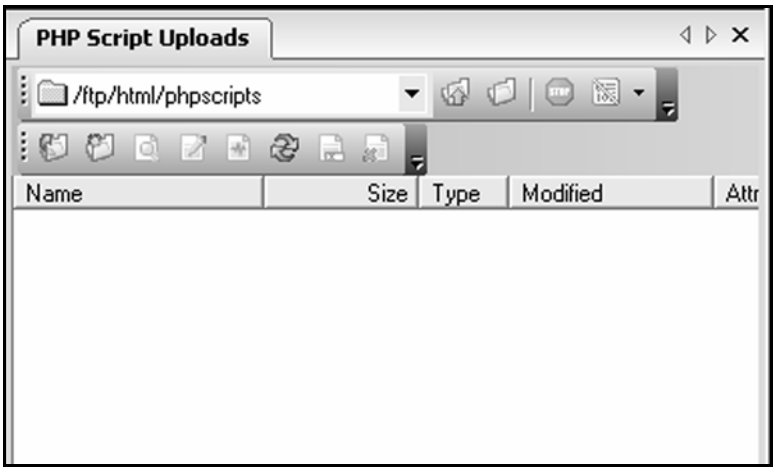


4. In the right-hand **PHP Script Uploads** pane, navigate to the **phpscripts** directory in your Web site.

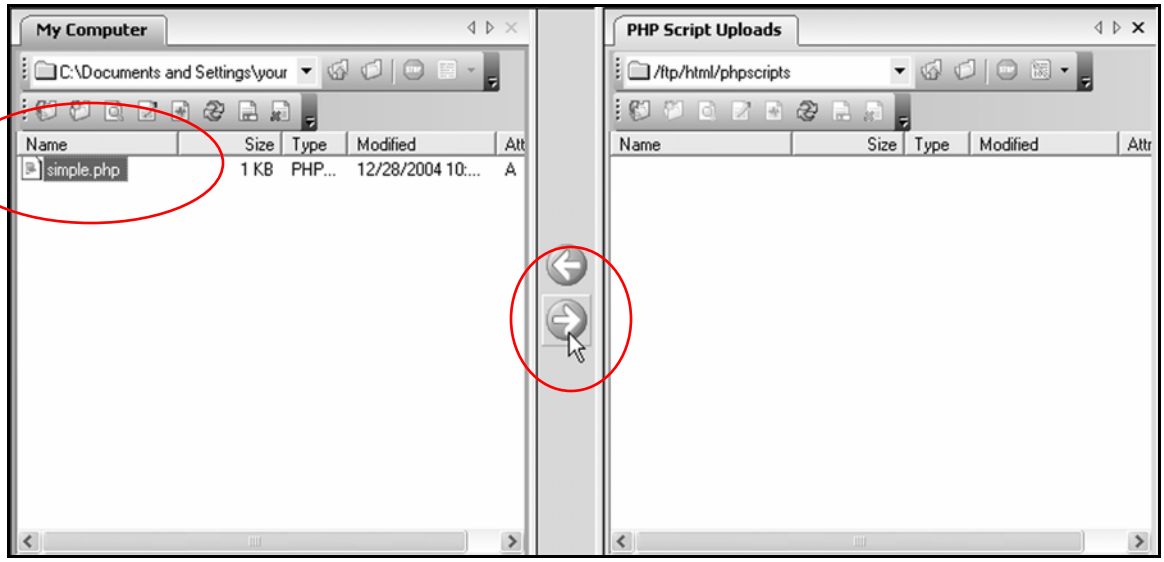


5. Double-click the **phpscripts** directory.

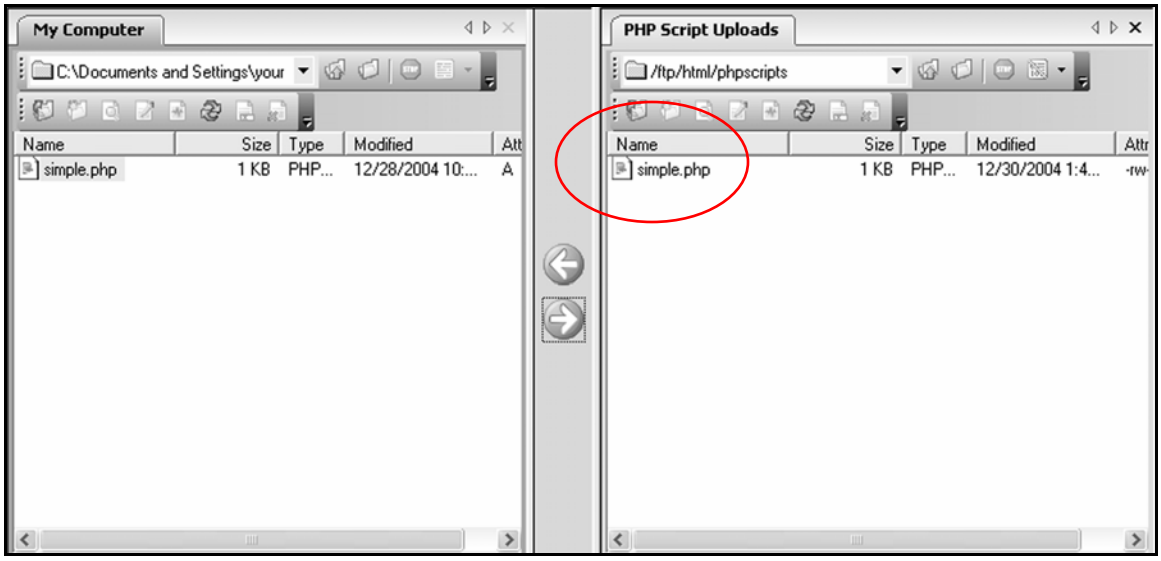
The pane should be blank:



6. Click **simple.php** in the **My Computer** pane, then click the button.



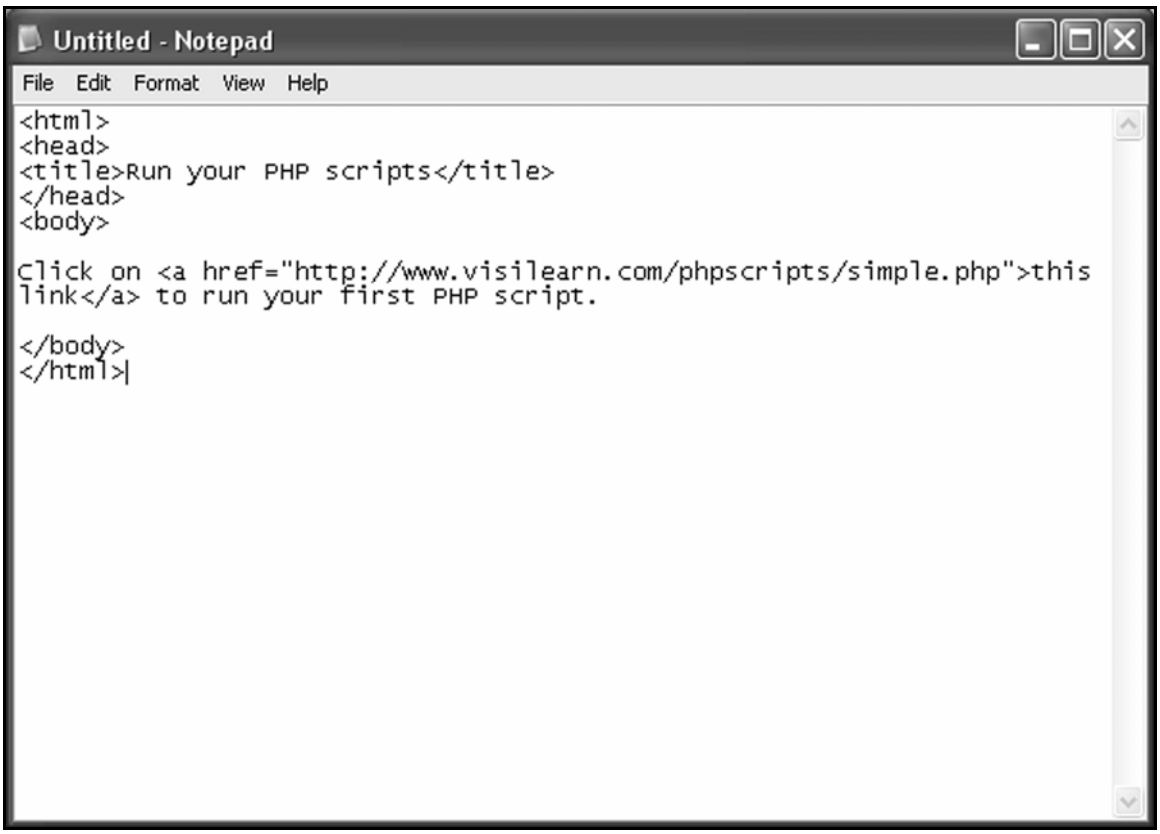
simple.php should now appear in the **PHP Script Uploads** pane:



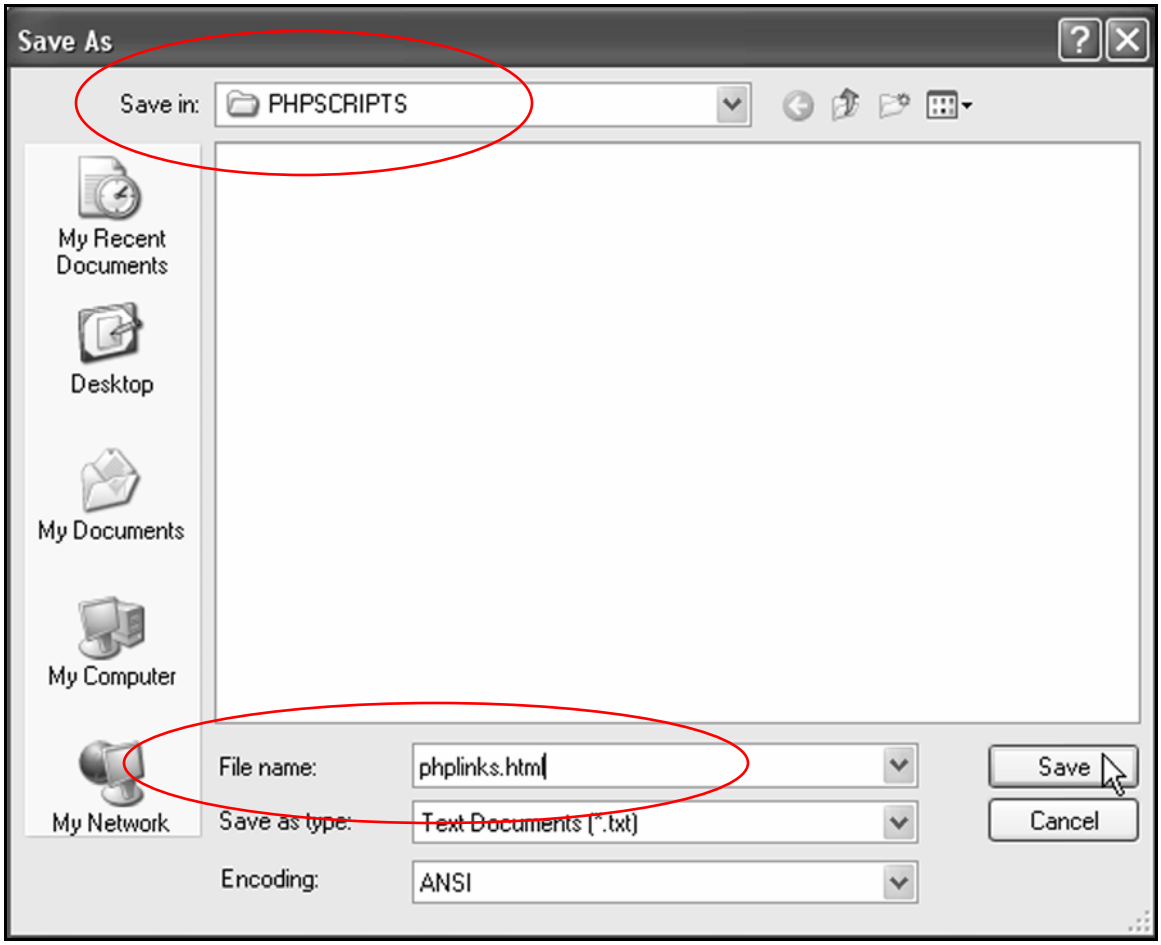
Run a script from a Web page

1. Using Notepad, create a new Web page with this code:

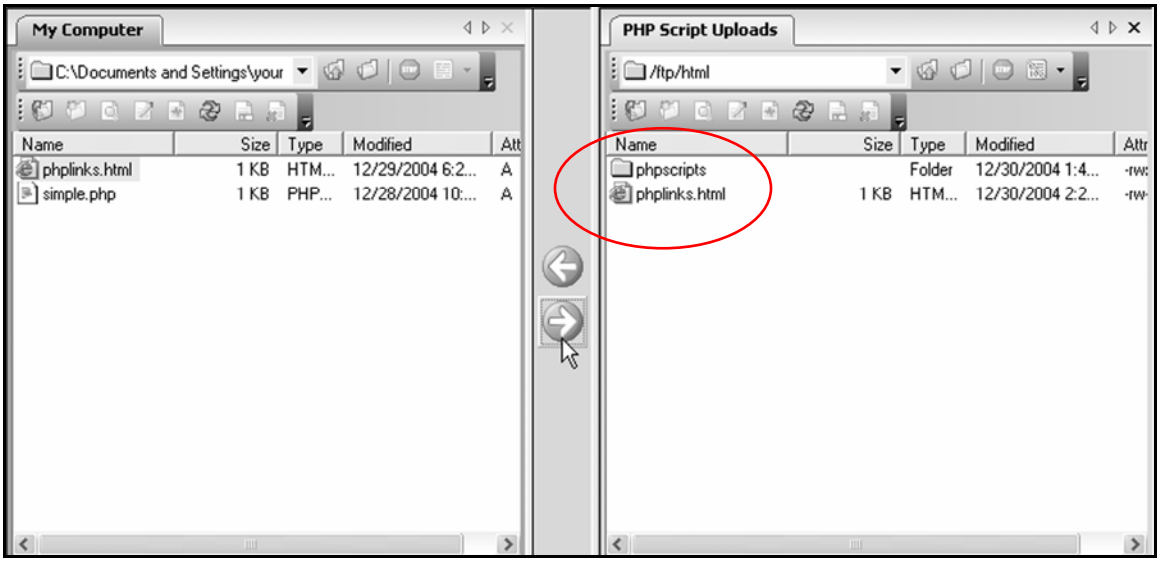
```
<html>
<head>
<title>Run your PHP scripts</title>
</head>
<body>

Click on <a
href="http://www.yourwebsite.com/phpscripts/sim
ple.php">this link</a> to run your first PHP
script.

</body>
</html>
```



2. Save the Web page as **phplinks.html** in the **PHPSCRIPTS** folder on your computer.



3. In WS_FTP, upload **phplinks.html** into the home directory of your Web site.



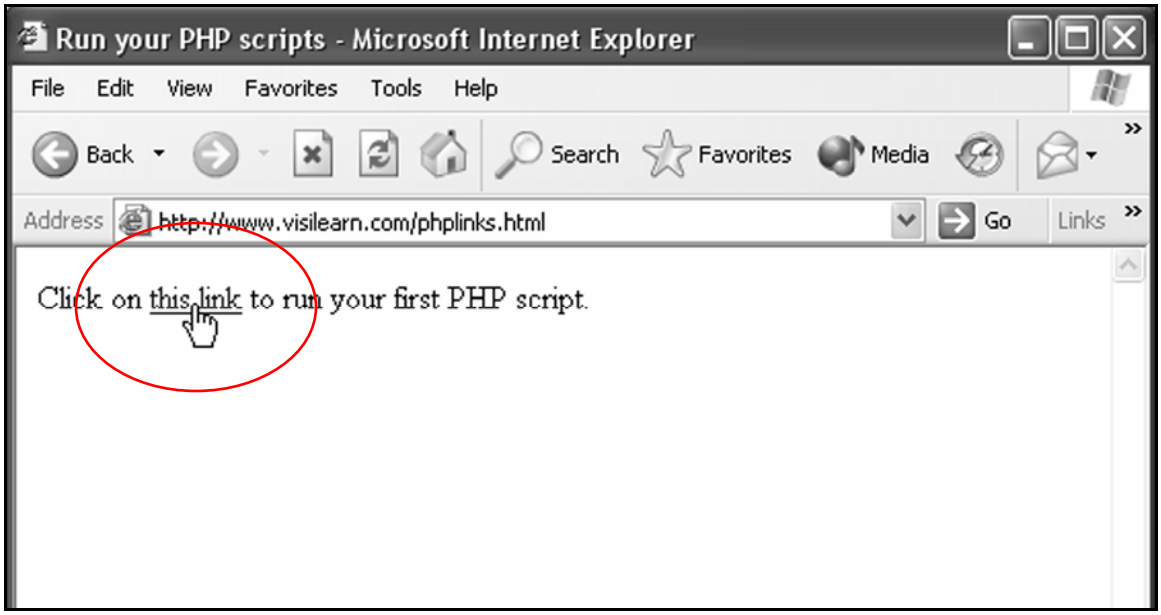
Tip: *Don't upload* **phplinks.html** *into the* **phpscripts** *directory.*

*Put it in the home directory of your Web site, where the home page—***index.html***—resides.*

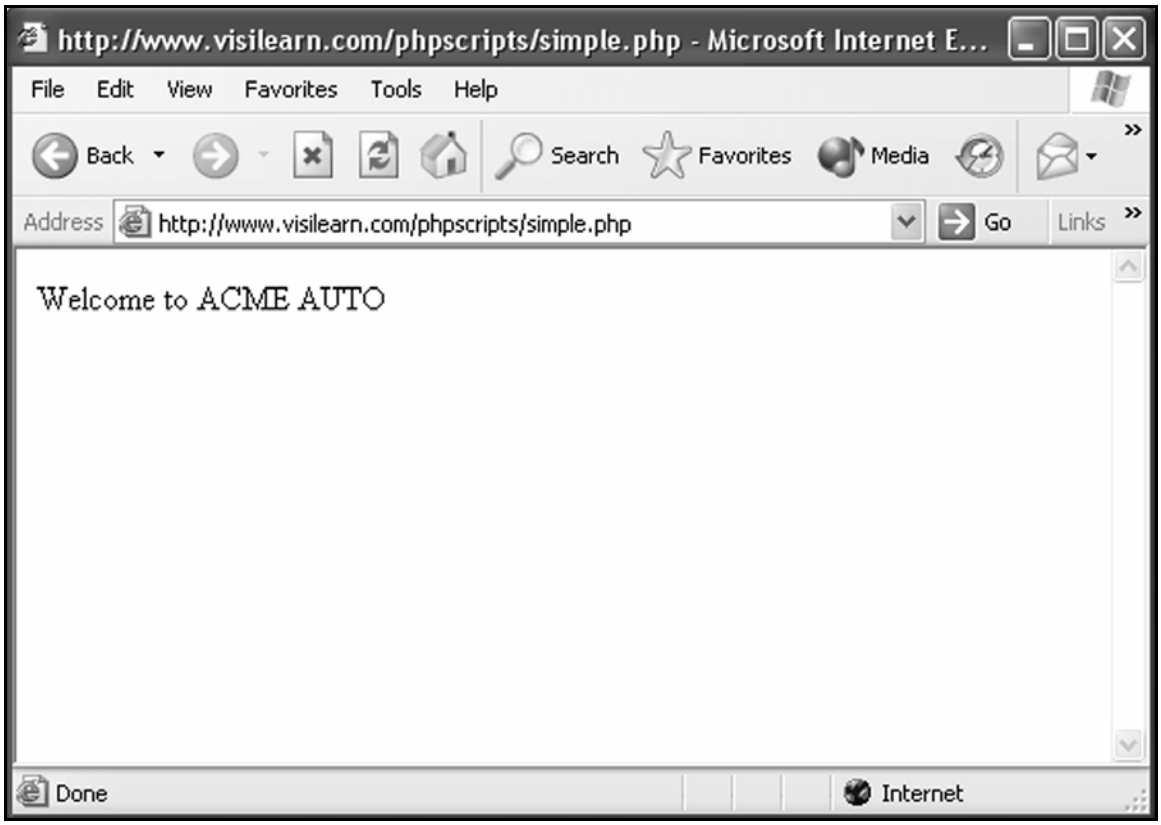
4. Open the Web browser and go to:

www.yourwebsite.com/phplinks.html

5. Click the link.



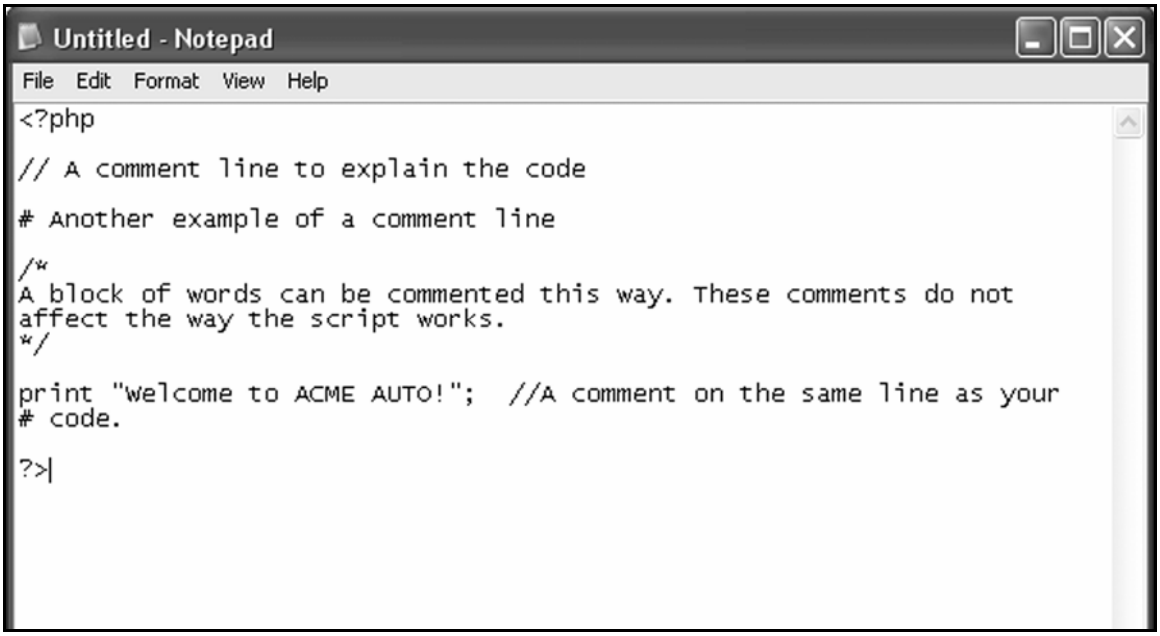
The output should look like this:



Insert comments

1. Using Notepad, create a new script with this code:

```
<?php

// A comment line to explain the code

# Another example of a comment line

/*
A block of words can be commented this way.
These comments do not affect the way
the script works.
*/

print "Welcome to ACME AUTO!";  // A comment on
the same line as your code.

?>
```



Tip: *If you're writing a comment in a script and it wraps to the next line, it needs a new # character in front.*

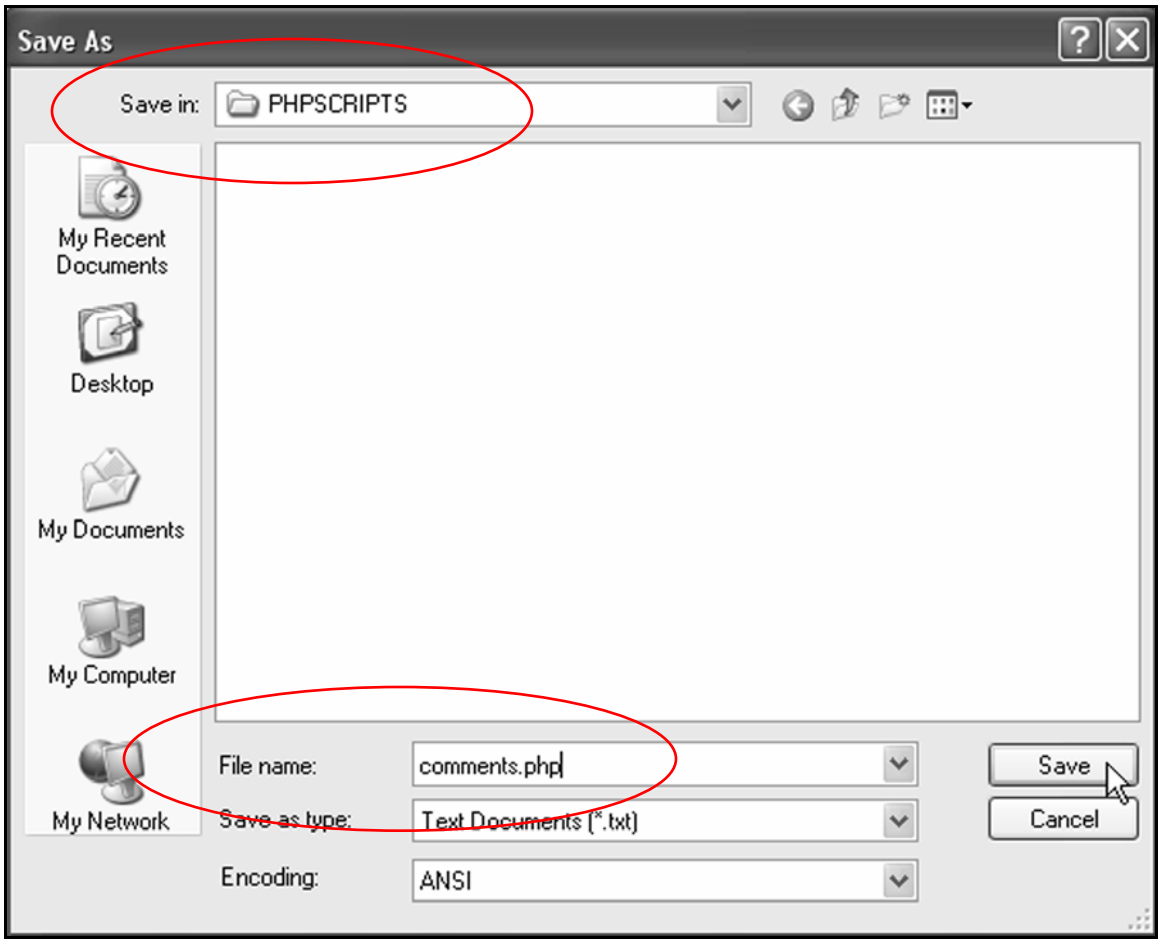
Incorrect:

```
# The second line lets a browser
display the script output.
```

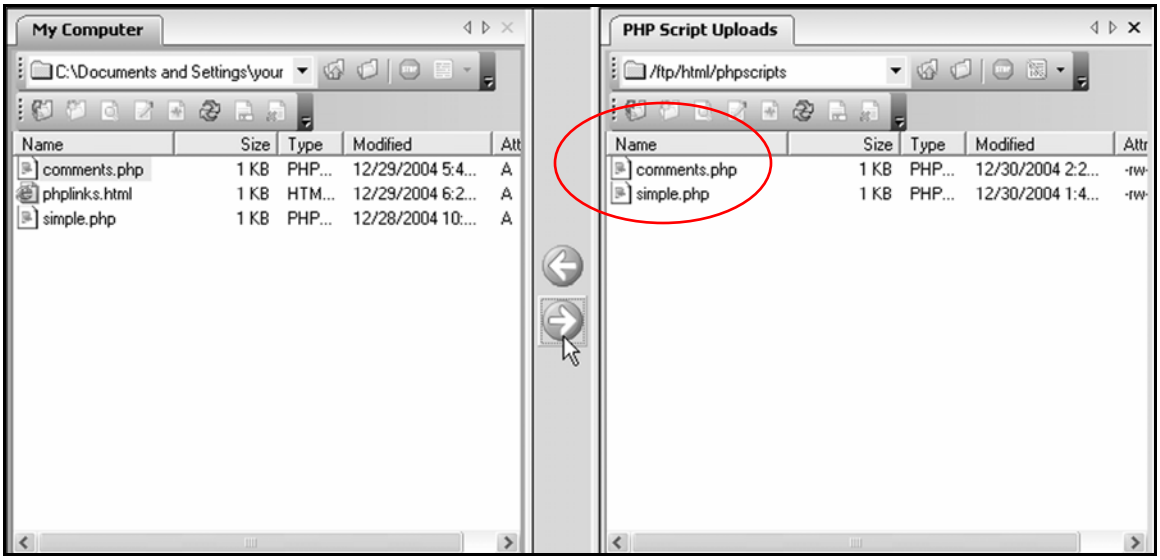
Correct:

```
# The second line lets a browser
# display the script output.
```

2. Save this script as **comments.php** in the **PHPSCRIPTS** folder on your computer.



3. Open WS_FTP and upload the **comments.php** script to the **phpscripts** directory in your Web site.



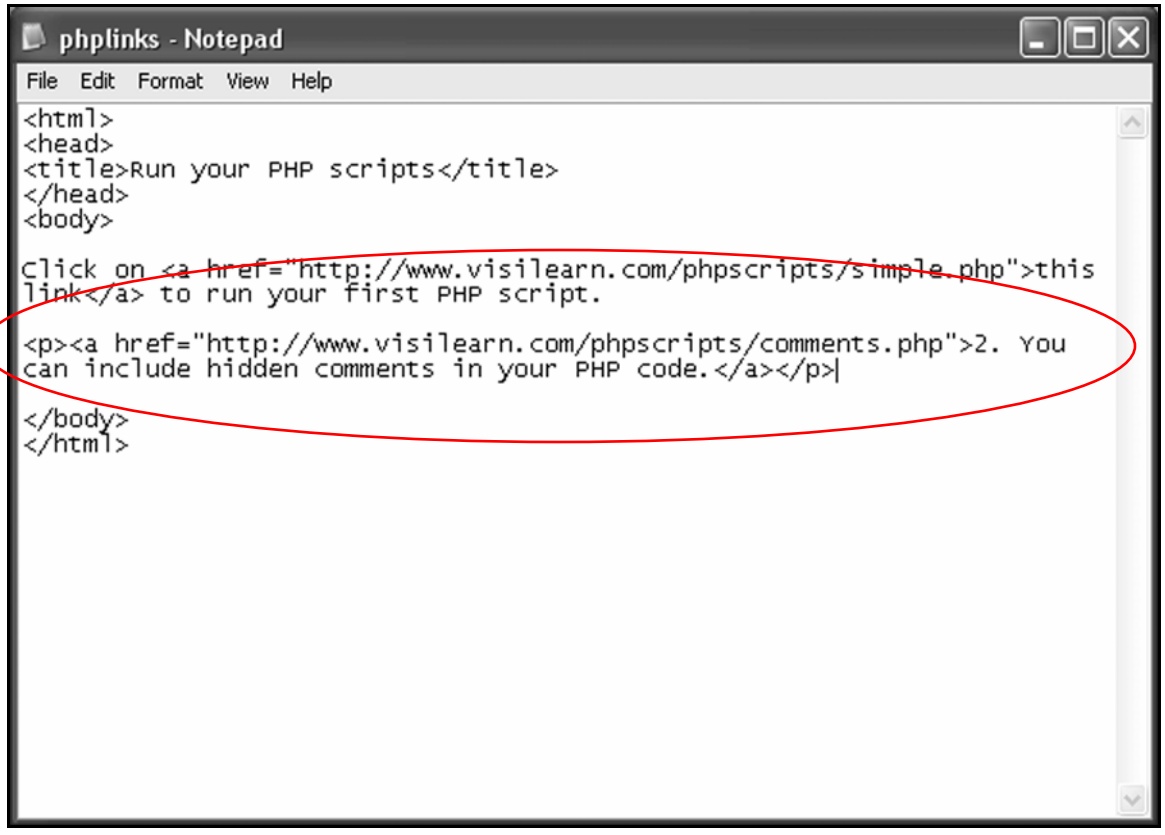
4. Open **phplinks.html** in Notepad.

Tip: *It's in the* **PHPSCRIPTS** *folder.*

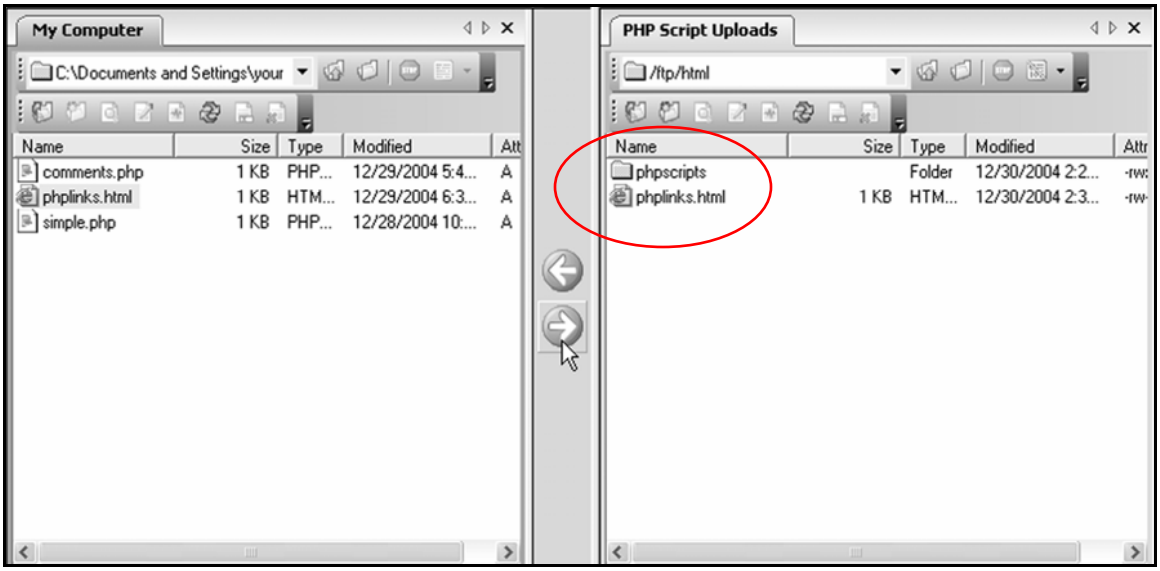
You may need to select **All Files** *in the* **Files of type** *list.*

5. Add a link to see the output of **format.php**:

```
<html>
<head>
<title>Run your PHP scripts</title>
</head>
<body>

Click on <a
href="http://www.yourwebsite.com/PHPSCRIPTS/sim
ple.php">this link</a> to run your first PHP
script.

<p><a
href="http://www.yourwebsite.com/phpscripts/com
ments.php">2. You can include hidden comments
in PHP code.</a></p>

</body>
</html>
```



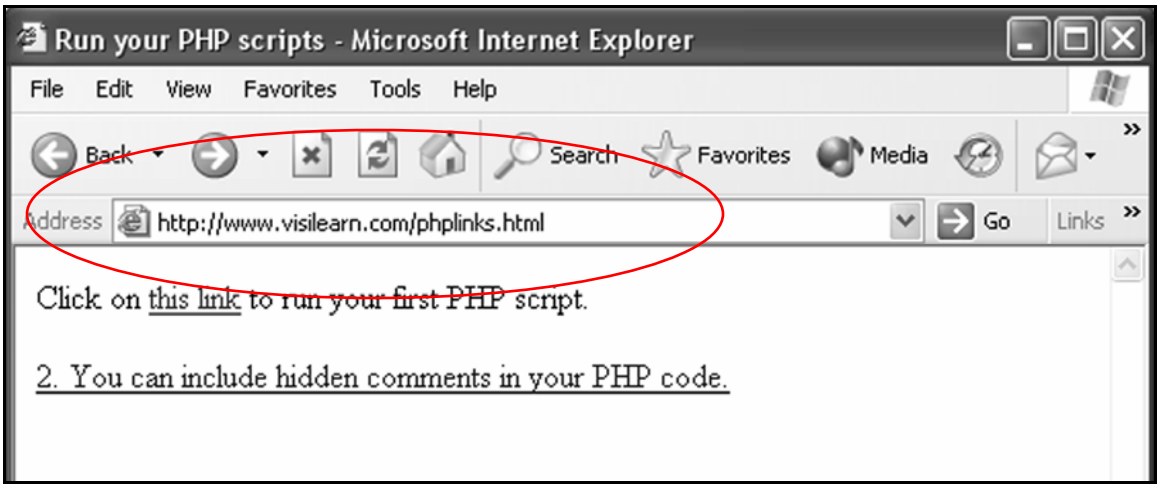
6. Save **phplinks.html**, then use WS_FTP to upload it to the home directory in your Web site.



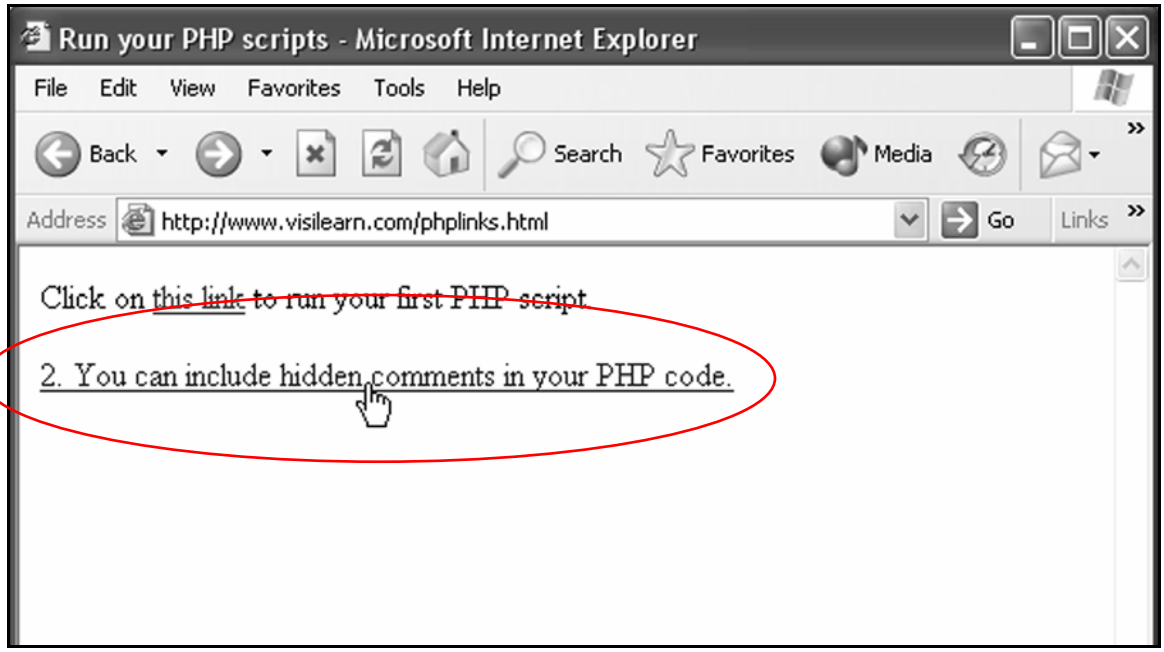
Tip: *This is the same place* **phplinks.html** *was before. When WS_FTP prompts you to replace the existing file, click the* Overwrite *button.*

7. Open the browser and go to:

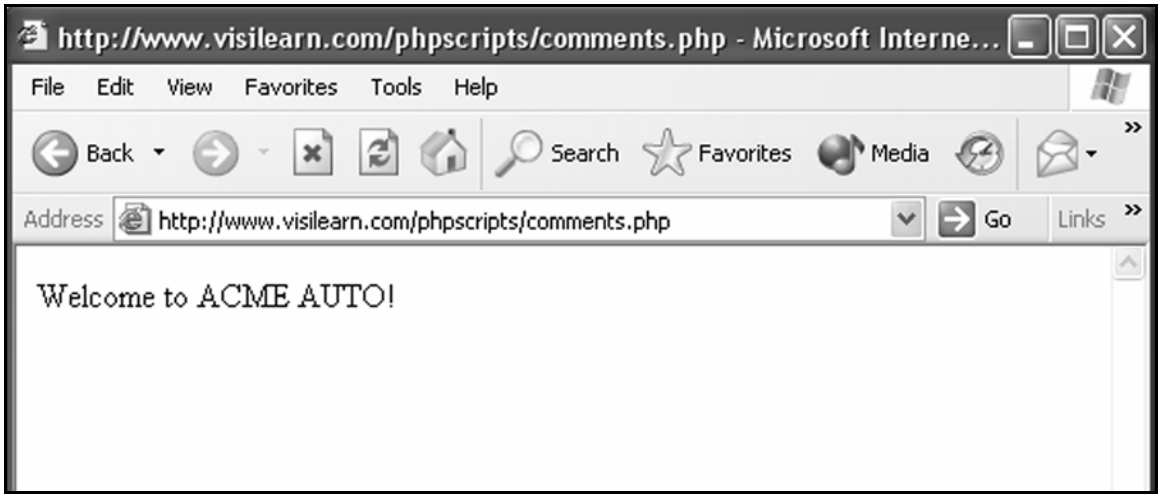
www.yourwebsite.com/ phplinks.html



8. Click the **2. You can include hidden comments in PHP code** link.



The output should look like this:

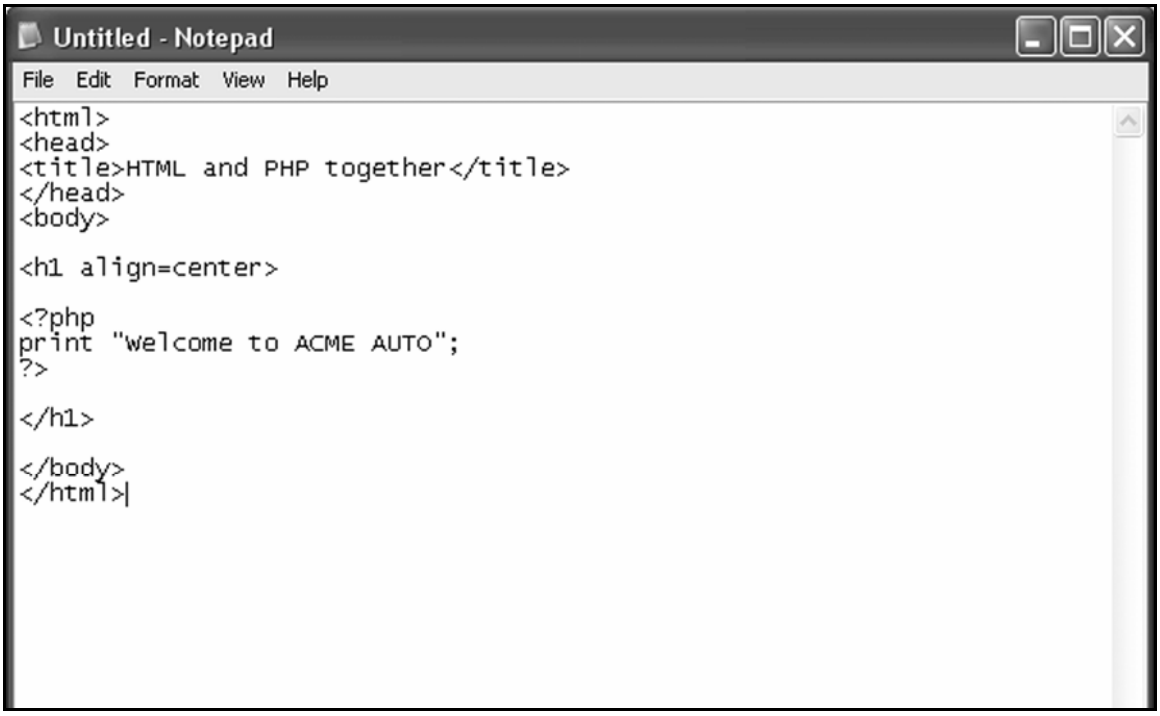


9. Close Notepad and WS_FTP.

Format output with HTML

1. In Notepad, create a new script with this code:

```
<html>
<head>
<title>HTML and PHP together</title>
</head>
<body>

<h1 align=center>

<?php
print "Welcome to ACME AUTO";
?>

</h1>

</body>
</html>
```

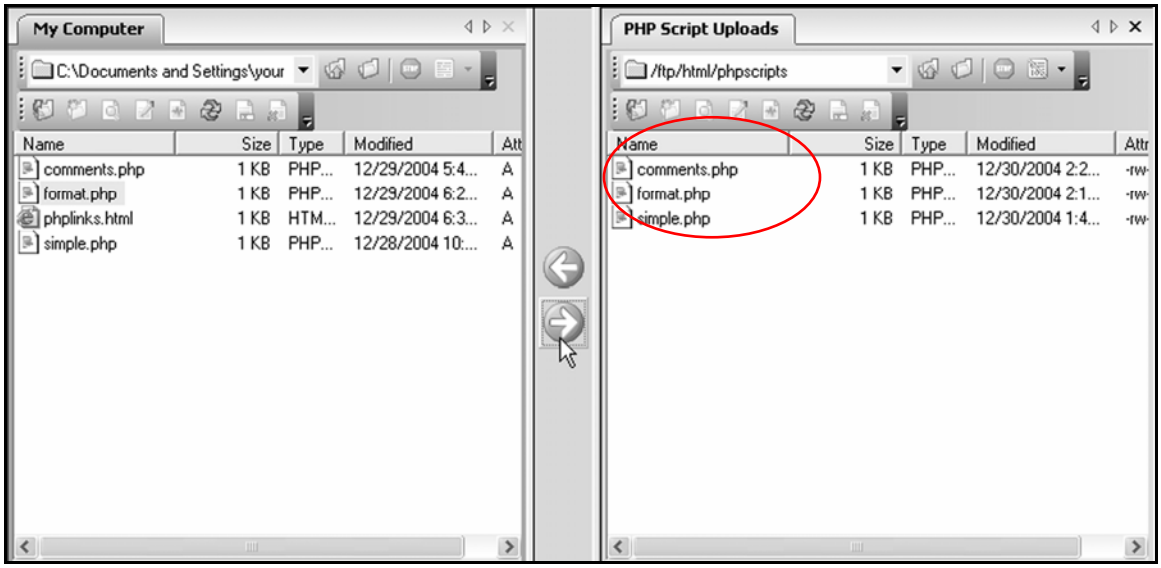


2. Save the script as **format.php** in the **PHPSCRIPTS** folder.

Tip: *This PHP script includes HTML tags that format the text it outputs.*

Notice that the `<h1 align=center>` *tag is outside of the PHP code. It's standard HTML.*

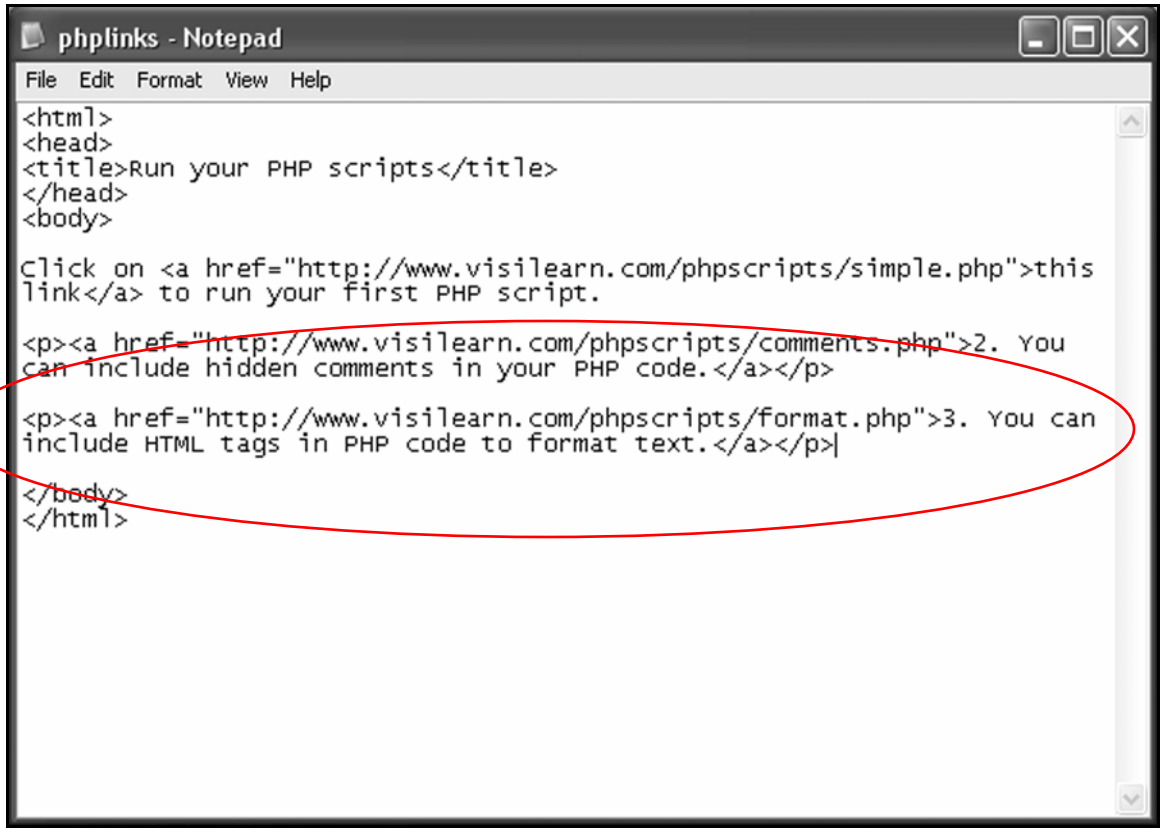
3. Upload **format.php** to the **phpscripts** directory in your Web site.



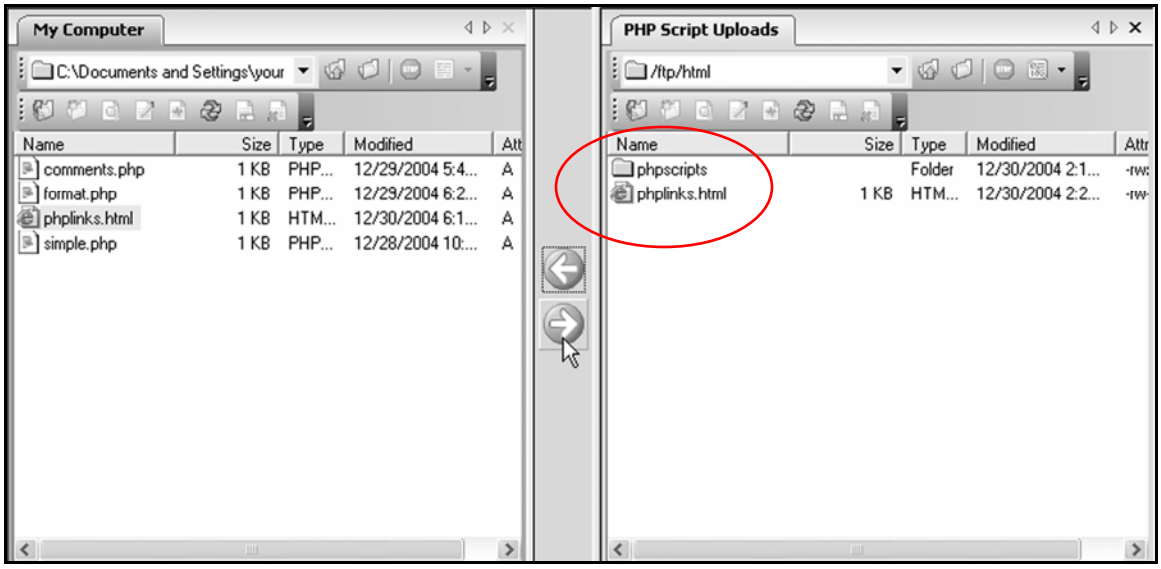
4. Open **phplinks.html** in Notepad.

5. Add a link to see the output of **format.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/com
ments.php">2. You can include hidden comments
in PHP code.</a></p>

<p><a
href="http://www.yourwebsite.com/phpscripts/for
mat.php">3. You can include HTML tags in PHP
code to format text.</a></p>
```

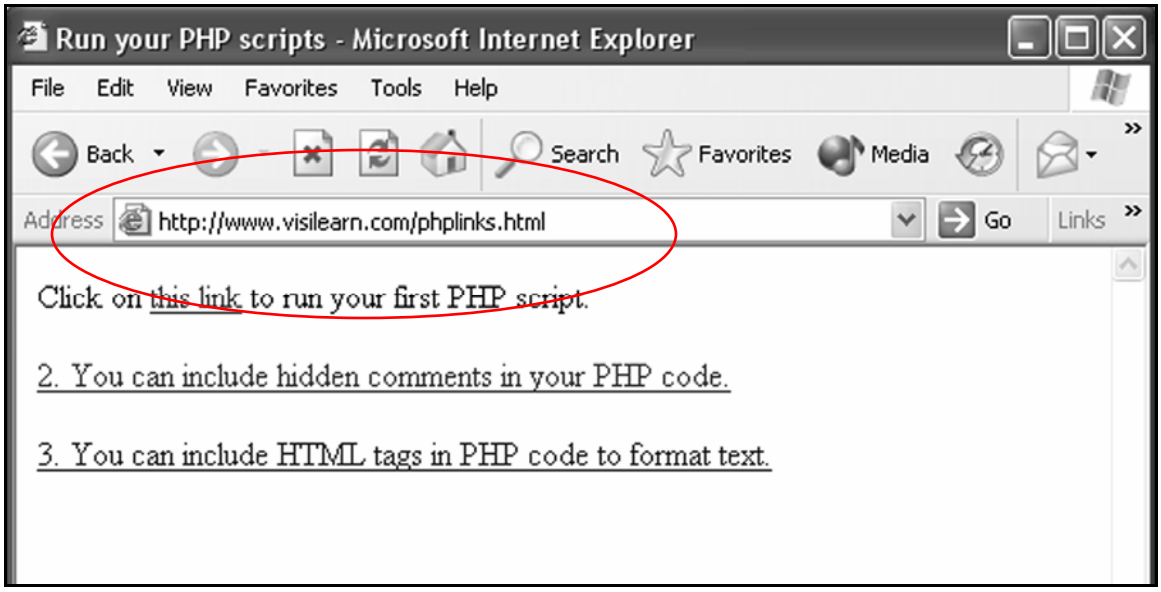


6. Save **phplinks.html**, then use WS_FTP to upload it to the home directory in your Web site.

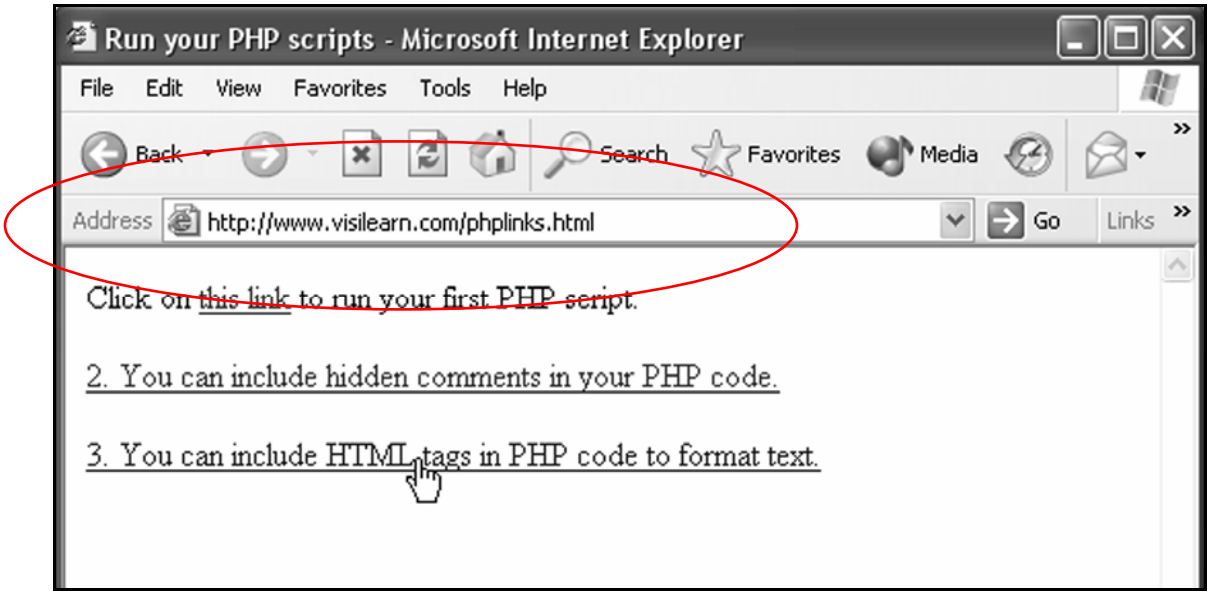


7. Open the browser and go to:

www.yourwebsite.com/phplinks.html



8. Click the **3. You can include HTML tags in PHP code to format text** link.



The output should look like this:

9. Close Notepad and WS_FTP.

Practice: Learning the Basics

1. Create a new folder called **PRACTICEPHP** on your computer's hard drive.

2. Open the Notepad program and create a new PHP script called **cars.php**.

   Write the script so it prints:

   Fast cars, vintage cars, and classic cars, we all have our favorite car.

   in a Web browser window.

3. Save **cars.php** in the **PRACTICEPHP** folder on your computer.

4. Open WS_FTP and create a new directory called **practicephp** directory in your Web site.

5. Upload **cars.php** to the **practicephp** directory in your Web site.

6. Using Notepad, create a new Web page called **practice.html** that contains a link to the PHP script **cars.php**:

   ```
   <p><a
   href="http://www.yourwebsite.com/practicephp/ca
   rs.php">Read about cars</a></p>
   ```

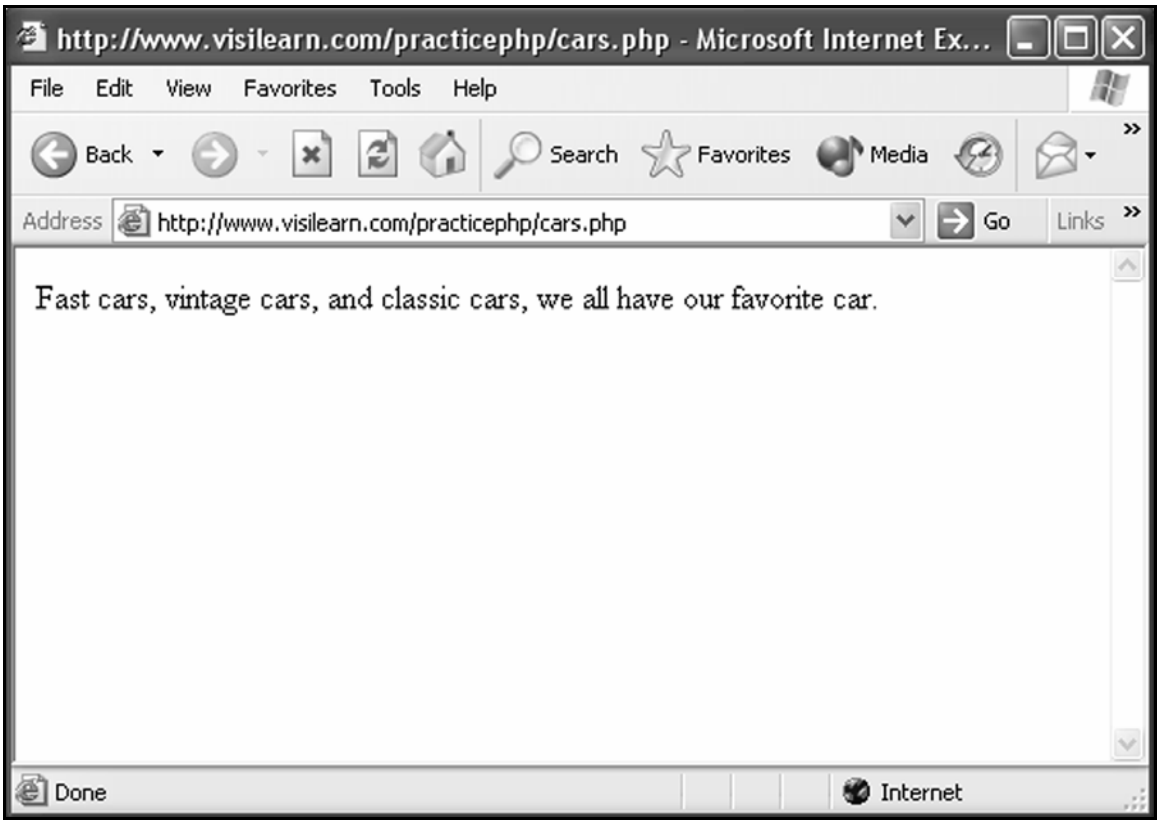
7. Save **practice.html**, then upload it to the home directory in your Web site.

8. In the browser, go to:

www.yourwebsite.com/practice.html

and click the **Read about cars** link.

The browser window should look like this:



9. Close Notepad and WS_FTP.

Working with Variables

In this section, you'll learn how to:

- **Employ single variables**
- **Print quotation marks**
- **Employ a list of variables**

What's a variable?

A variable is a placeholder for information within a PHP script. Data is stored and retrieved using variables.

In PHP, a **variable** is can be a single piece of information, or a list of things store in an array. They always start with a dollar sign, and are case sensitive.

E*xample:* **`$myName`**

Variables are essential to all programming, and very useful. For example, you can use a variable to easily change "brown eyes" to "blue eyes" in a PHP script:

```
$eyecolor="brown";
```

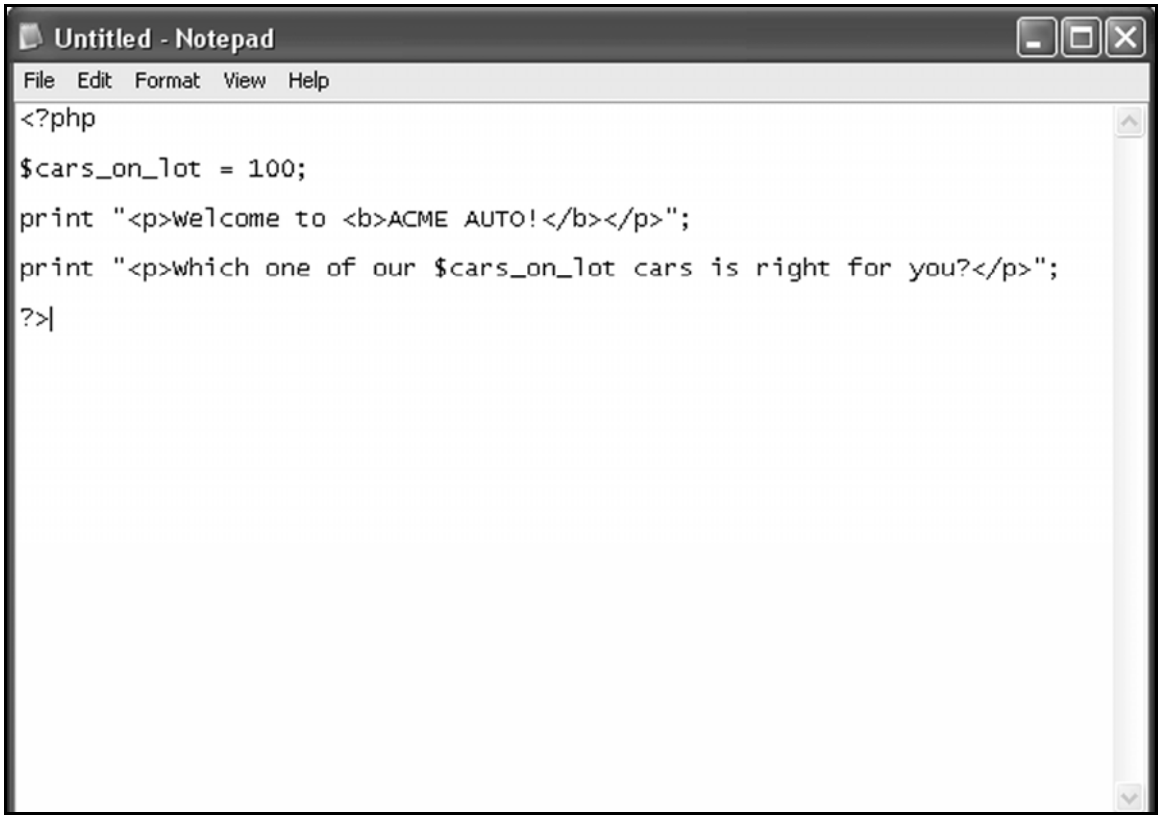
As the old song says, "Don't it make my $eyecolor eyes blue…"

Employ single variables

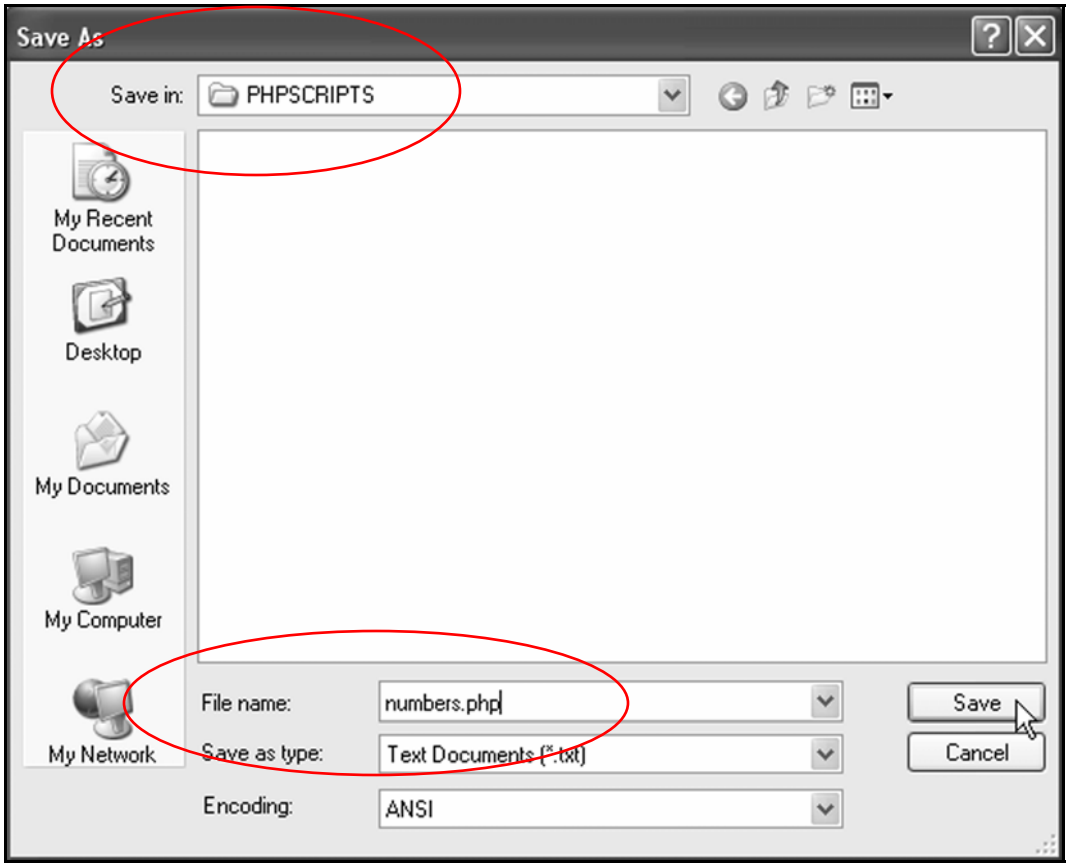
Assign a number to a variable

1. Open Notepad, then create a new script with this code:

```
<?php

$cars_on_lot = 100;

print "<p>Welcome to <b>ACME AUTO!</b></p>";

print "<p>Which one of our $cars_on_lot cars is
right for you?</p>";

?>
```



2. Save the script as **numbers.php** in the **PHPSCRIPTS** folder.



Here's what each line of the script does:

- **`<?php`**

  This is the starting PHP tag. Your PHP code starts after this tag.

- **`$cars_on_lot = 100;`**

  `$cars_on_lot` is the variable. Variables start with a **`$`**.

  The number 100 is assigned to the variable. The number is easy to change—that's why it's called a variable.

- **`print "<p>Welcome to <b>ACME AUTO!</b></p>";`**

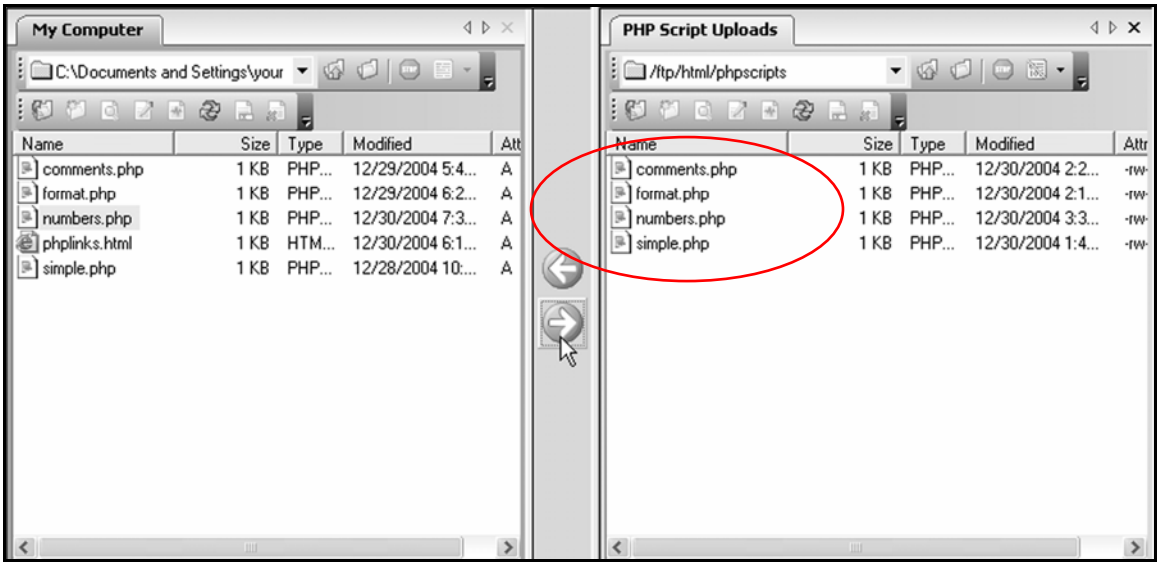
  `print "<p>Which one of our $cars_on_lot cars is right for you?</p>";`

  These lines should also look familiar. They're HTML code like we've used before, but with a difference: **`$cars_on_lot`**.

  This variable tells the Web browser to get the number specified (100) and insert it here.

  You'll see how it works in the following steps.

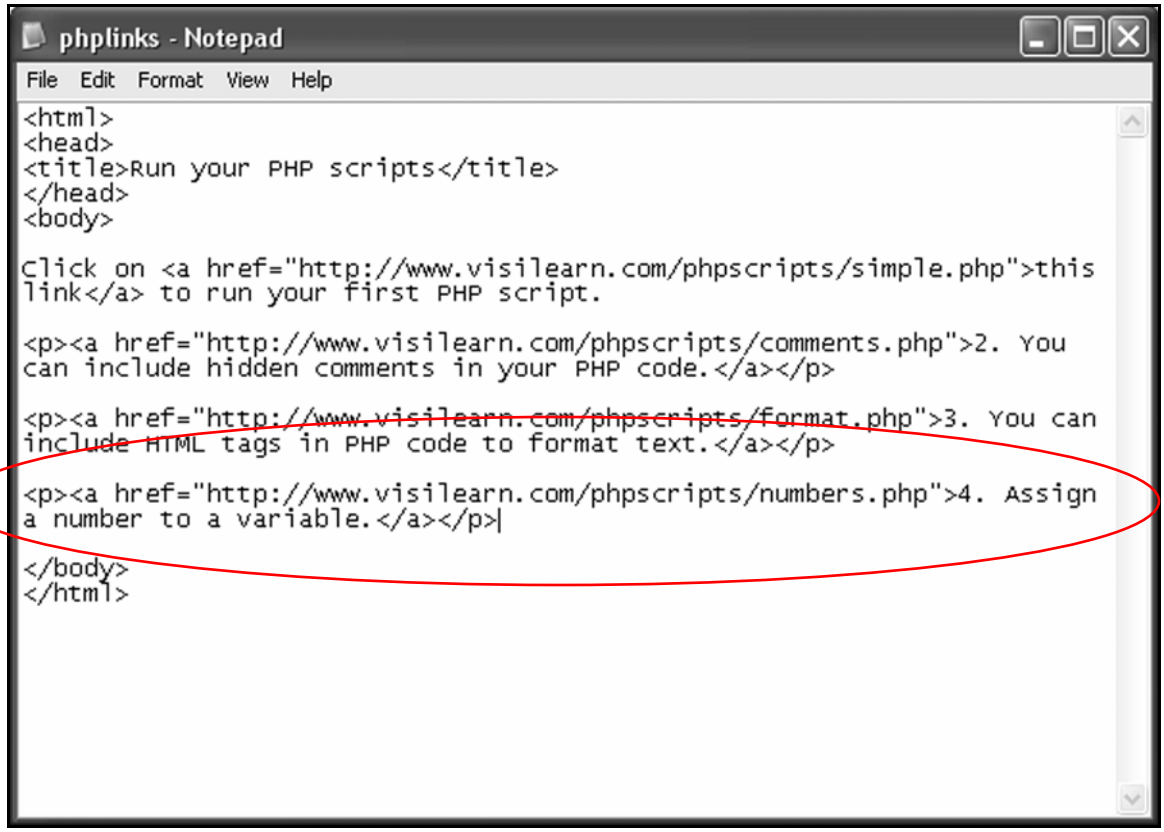
3. Open WS_FTP, then upload **numbers.php** to the **phpscripts** directory in your Web site.



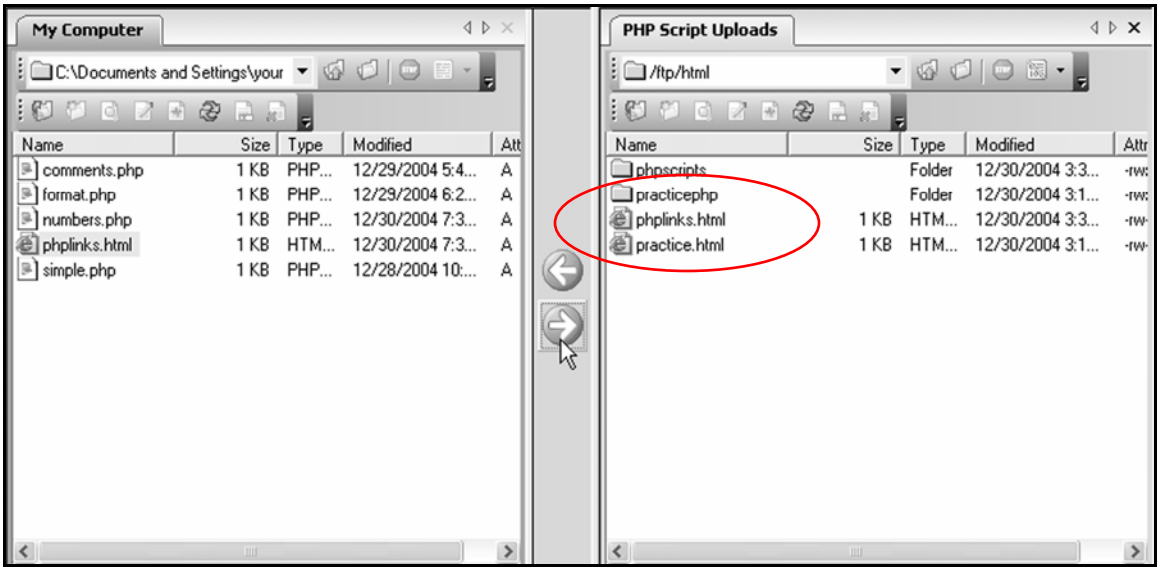
4. In Notepad, open **phplinks.html**.

5. Insert a new link to **numbers.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/for
mat.php">3. You can include HTML tags in PHP
code to format text.</a></p>

<p><a
href="http://www.yourwebsite.com/phpscripts/num
bers.php">4. Assign a number to a
variable.</a></p>
```



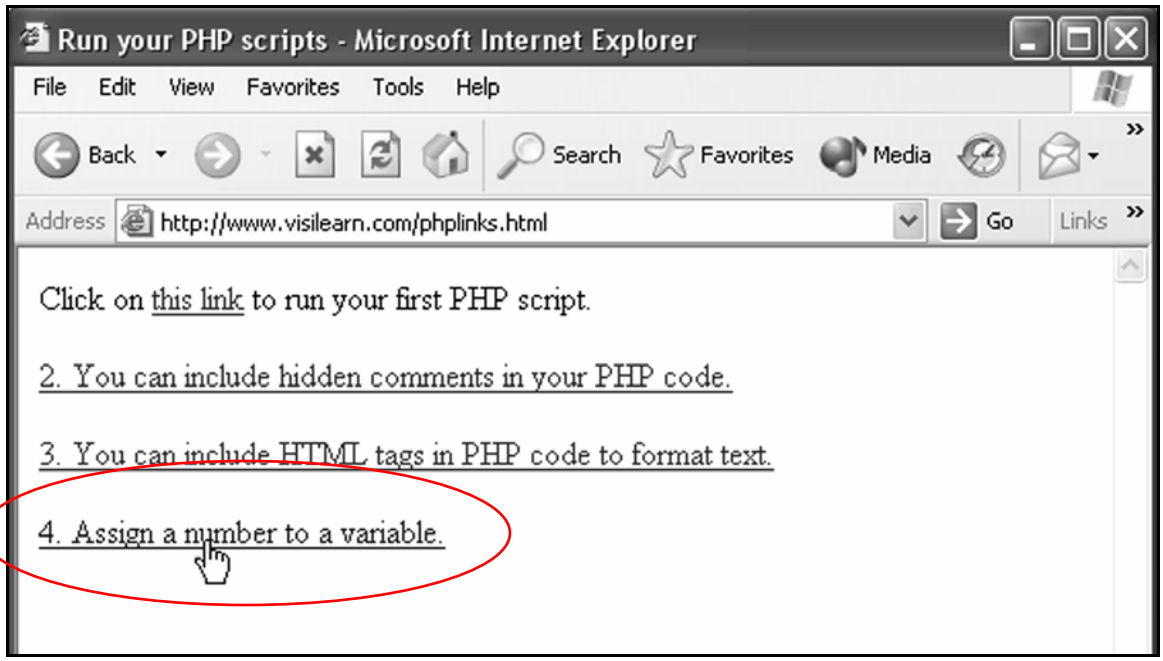
6. Save **phplinks.html**, then upload it to the home directory in your Web site.



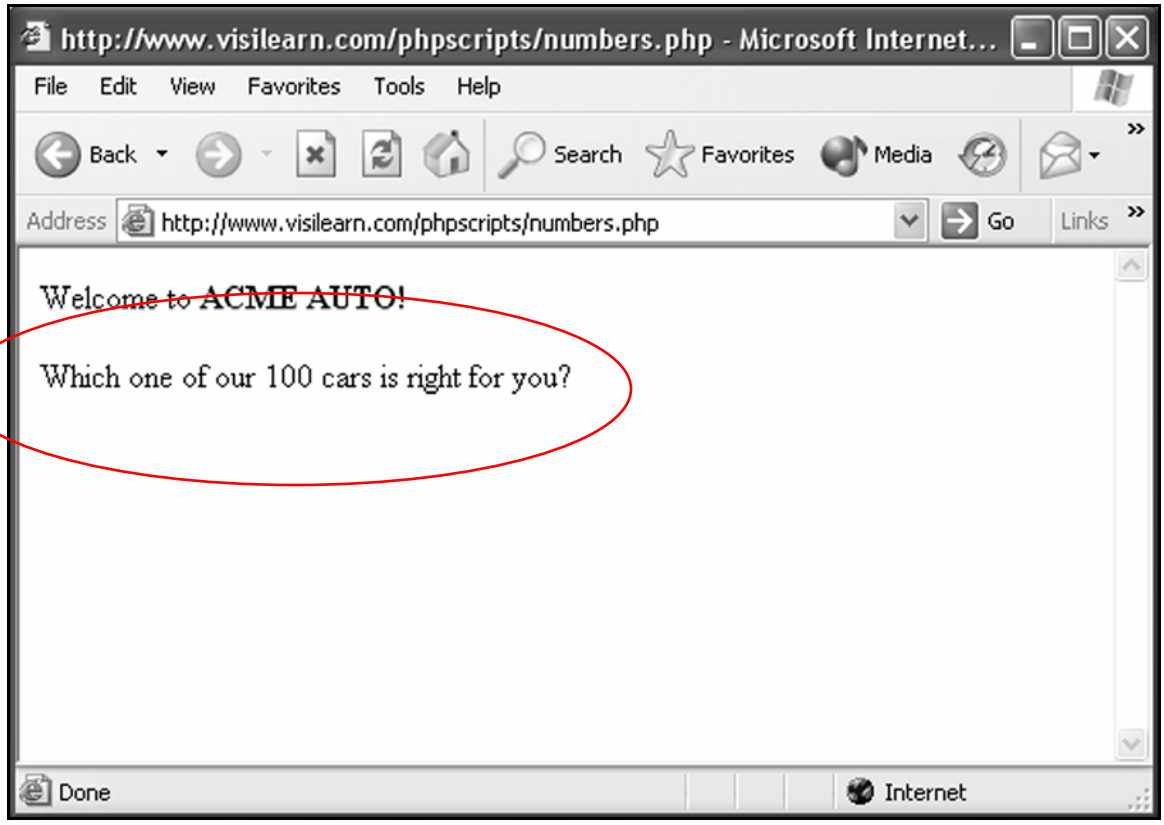
7. Using the browser, go to:

www.yourwebsite.com/phplinks.html

8. Click the **4. Assign a number to a variable** link.



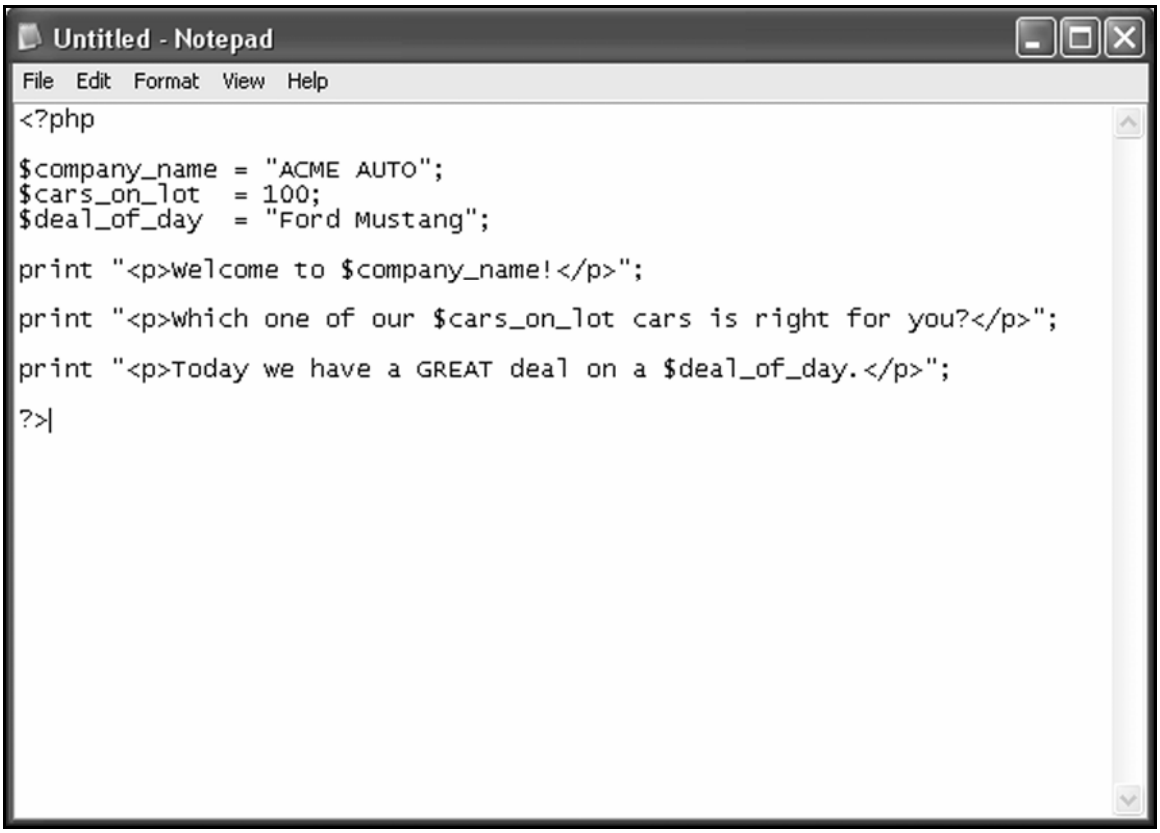
The output should look like this:



Assign text to a variable

1. Using Notepad, create a new script with this code:

```
<?php

$company_name = "ACME AUTO";
$cars_on_lot  = 100;
$deal_of_day  = "Ford Mustang";

print "<p>Welcome to $company_name!</p>";

print "<p>Which one of our $cars_on_lot cars is
right for you?</p>";

print "<p>Today we have a GREAT deal on a
$deal_of_day.</p>";

?>
```



2. Save the script as **text.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- `$company_name = "ACME AUTO";`

  Assigns the text `ACME AUTO` to the variable `$company_name`.

- `$deal_of_day  = "Ford Mustang";`

  Assigns the text Ford Mustang to the variable `$deal_of_day`.

- `$cars_on_lot  = 100;`

  Assigns the number `100` to the variable `$cars_on_lot`.

- `print "<p>Welcome to $company_name!</p>\n";`

  Prints the words "**Welcome to**" to the browser window, then inserts the text assigned to the variable `$company_name` ("ACME AUTO").

- `print "<p>Which one of our $cars_on_lot cars is right for you?</p>\n";`

  Prints words to the browser window, inserting the number assigned to the variable `$cars_on_lot` (100).

- `print "<p>Today we have a GREAT deal on a $deal_of_day.</p>\n";`

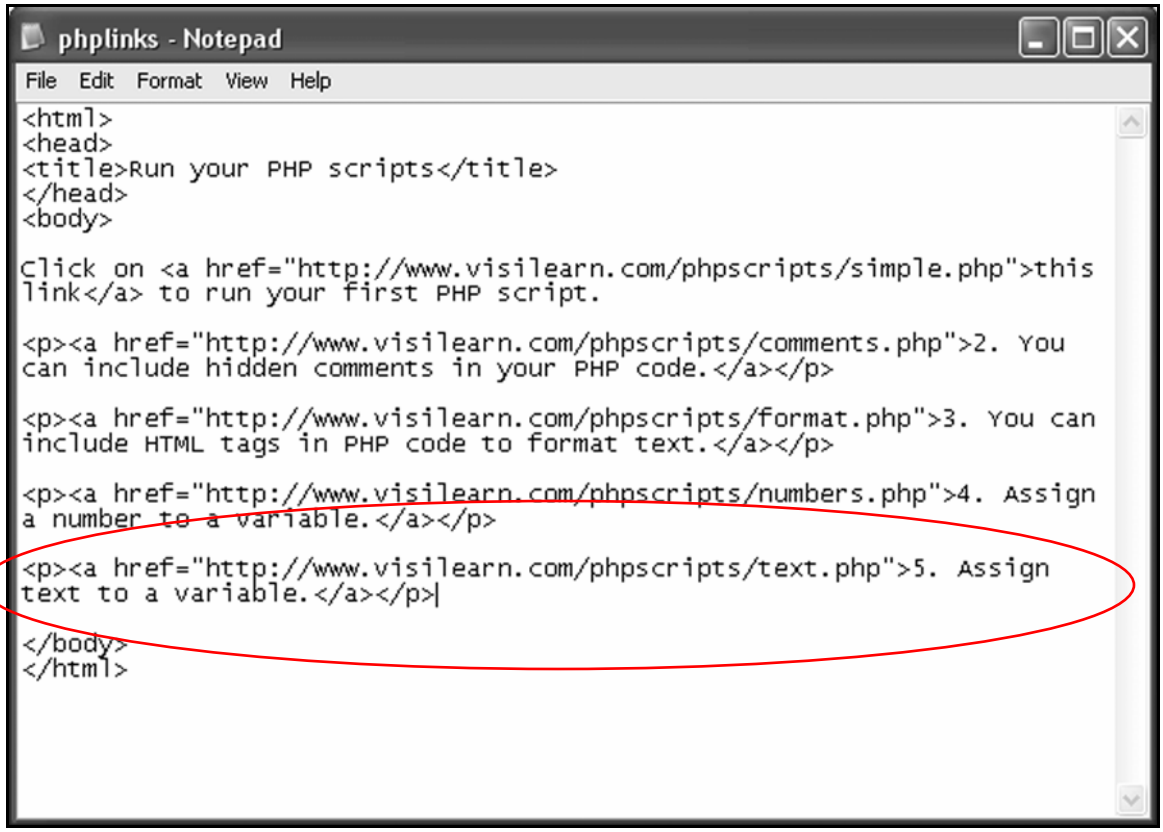
  Prints words to the browser window, then inserts the text assigned to the variable `$deal_of_day` ("Ford Mustang").

3. Upload text.php to the **PHPSCRIPTS** directory in your Web site.

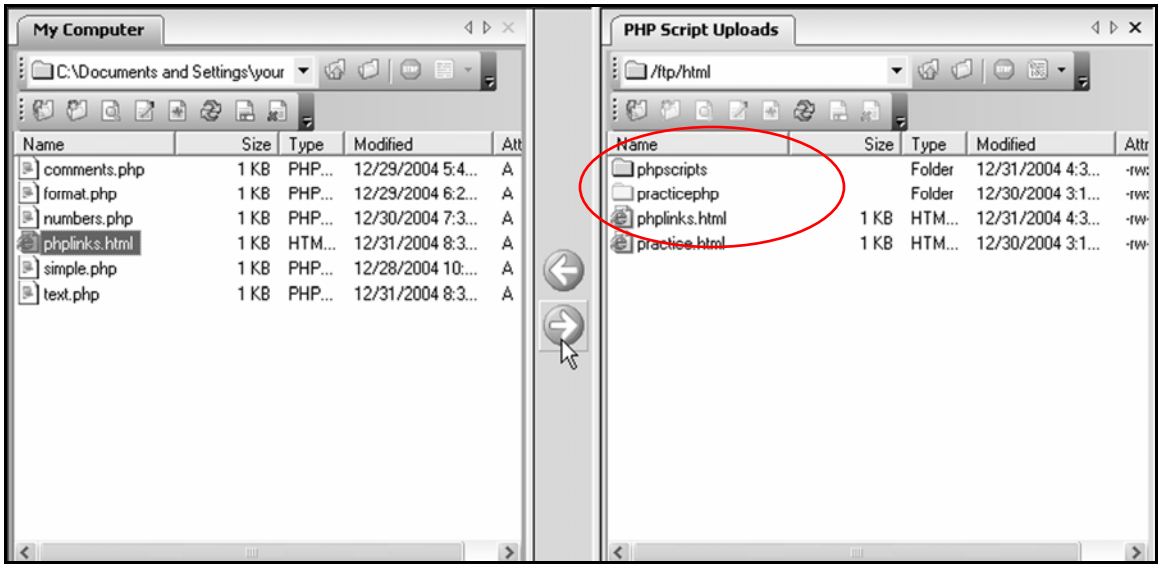
4. In Notepad, open **phplinks.html**.

5. Insert a new link to **text.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/tex
t.php">5. Assign text to a variable.</a></p>
```



6. Save **phplinks.html**, then upload it to the home directory in your Web site.

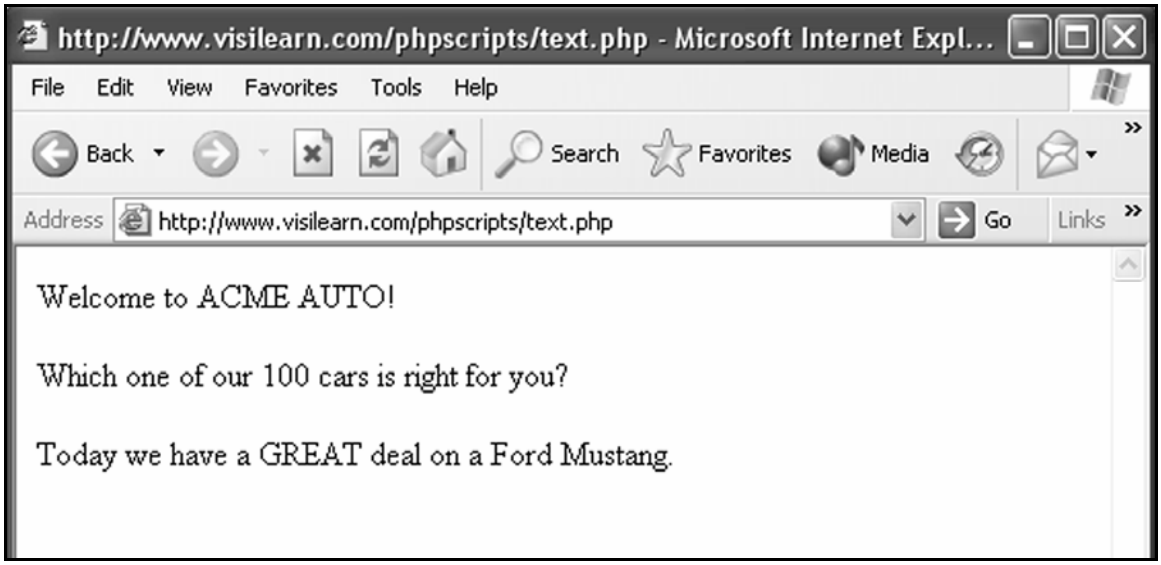


7. Using the browser, go to:

www.yourwebsite.com/phplinks.html

8. Click the **5. Assign text to a variable** link.

The output should look like this:

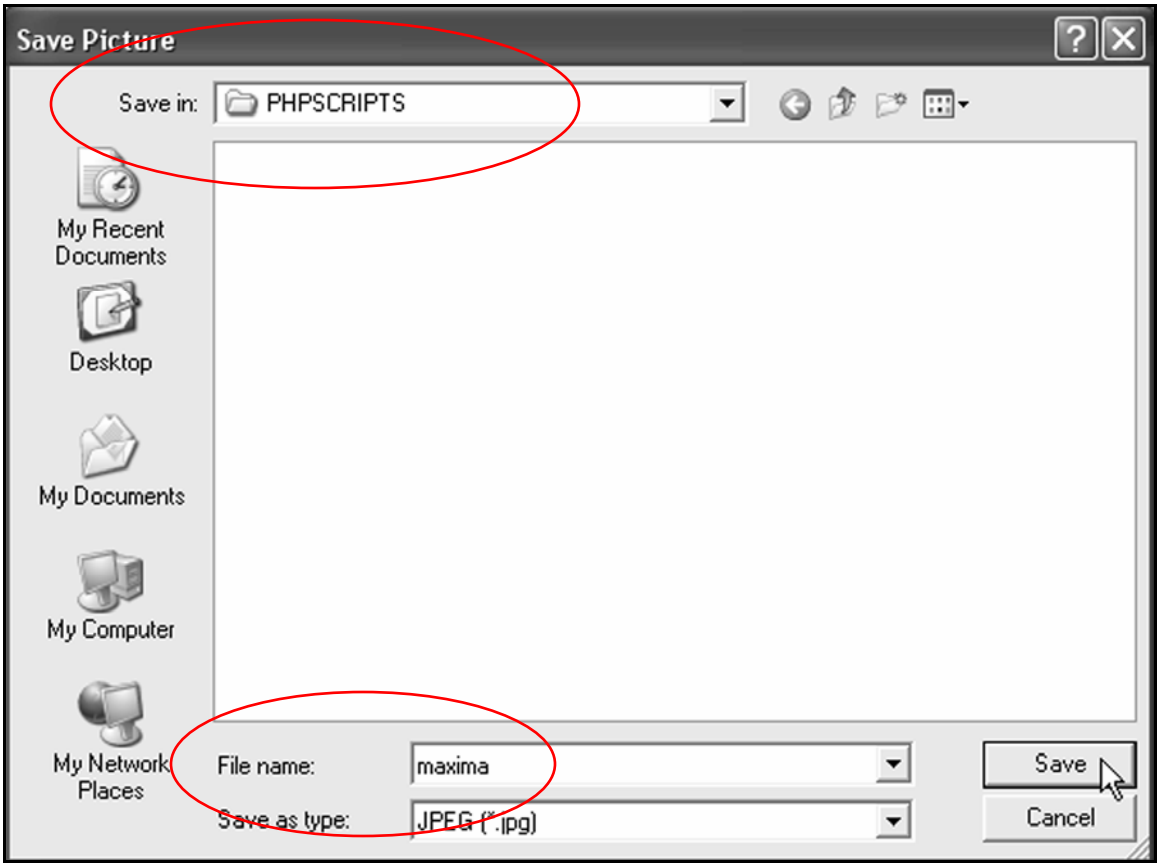


Print quotation marks

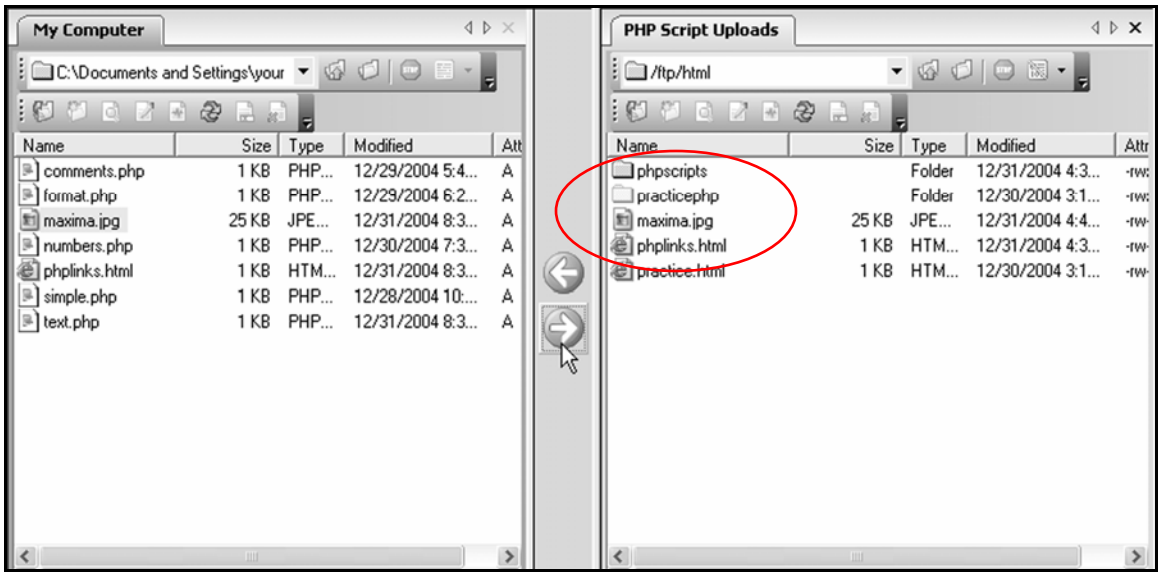
1. In the browser, go to:

   www.visibooks.com/books/php

2. Right-click **maxima.jpg**, then save it in the **PHPSCRIPTS** folder on your computer.



3. Upload **maxima.jpg** to the home directory in your Web site.



4. In Notepad, create a new script with this code:

```
<?php

$cars_on_lot = 100;

$deal_of_day = "Nissan Maxima";

$pic_of_day =
"http://www.yourwebsite.com/maxima.jpg";

print "<p>Welcome to <b>ACME AUTO!</b></p>";

print "<p>Which one of our $cars_on_lot cars is
right for you?</p>";

print "<p>Today we have a <b>GREAT</b> deal on
a $deal_of_day car:</p>";

print "<img src="$pic_of_day">";

?>
```

Tip: *Remember to change the* `www.yourwebsite.com` *address in* `$pic_of_day = "http://www.yourwebsite.com/maxima.jpg"` *to your actual Web site address.*

5. Save the script as **qmarks.php** in the **PHPSCRIPTS** folder.

6. Upload **qmarks.php** to the **phpscripts** directory in your Web site.

7. In Notepad, open **phplinks.html**.

8. Insert a new link to **qmarks.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/qma
rks.php">6. Print quotation marks.</a></p>
```

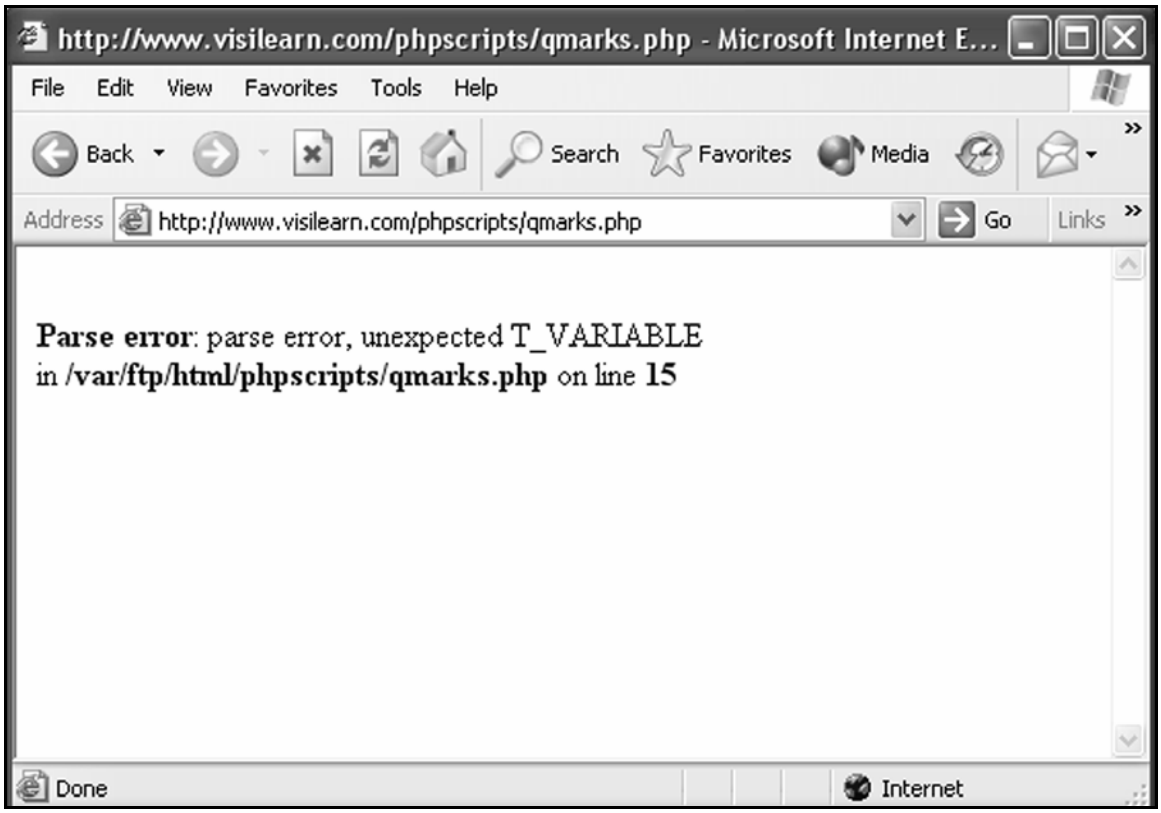
9. Save **phplinks.html**, then upload it to the home directory in your Web site.

10. Using the browser, go to:

www.yourwebsite.com/phplinks.html

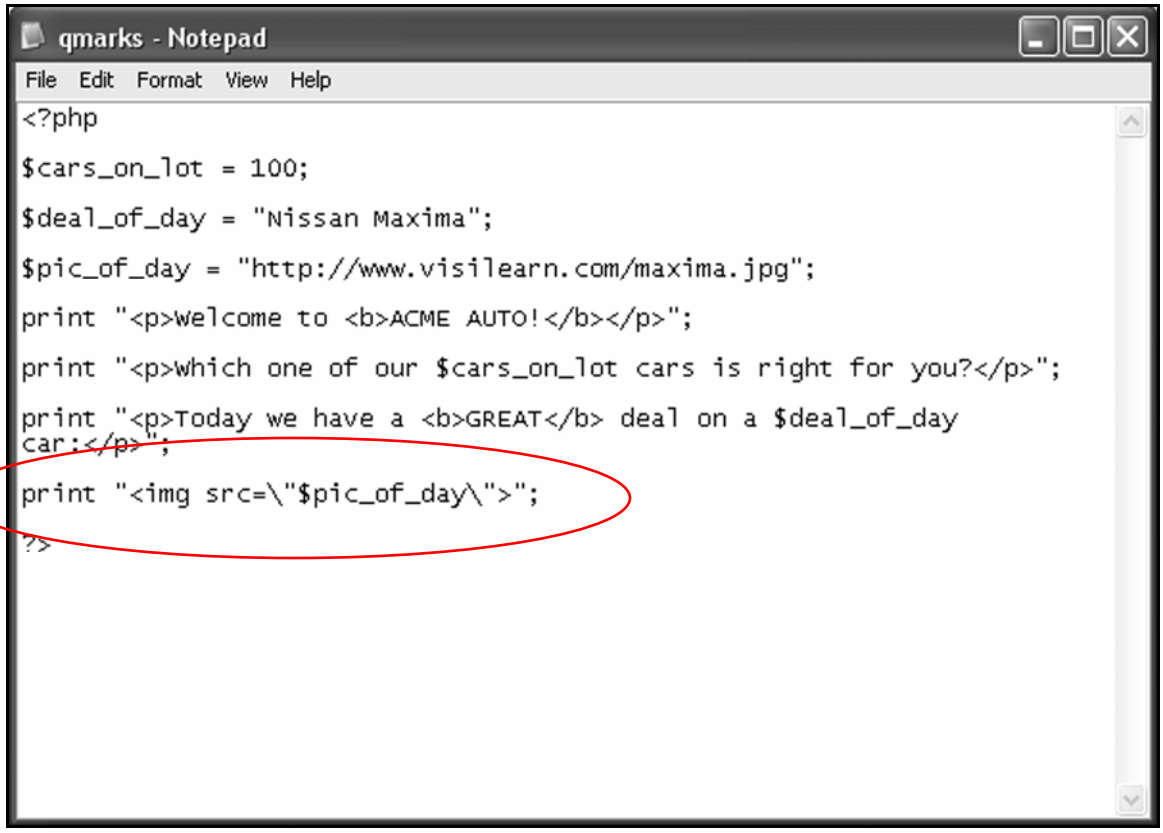
11. Click the **6. Print quotation marks** link.

The output should look something like this:



12. In Notepad, edit **qmarks.php** to enclose the `$pic_of_day` variable in \ characters:

```
print "<img src=\"$pic_of_day\">";
```

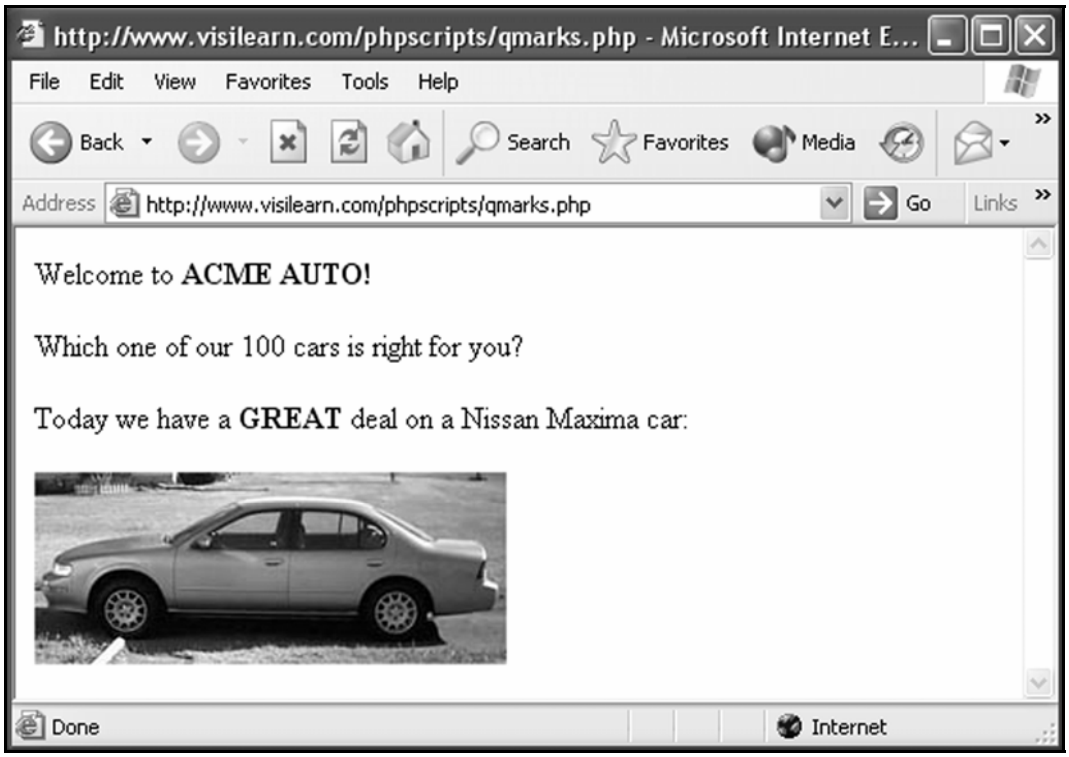


13. Save **qmarks.php** and upload it to the **phpscripts** directory again.

14. Reload **phplinks.html** in your Web browser.

15. Click the **Print quotation marks** link again.

Its output should look like this:



Tip: *Since the HTML* `<img>` *tag requires the use of two double-quotation marks inside the already quoted print statement.*

```
<img src="maxima.jpg">
```

enclose them in \ *characters to let the Web server know that you want to print a double-quote to the screen. Otherwise, the Web server will think you want the double-quotes to start and end a text string in a PHP command.*

\ *is called an "escape character." Escape characters are used to print characters, such as double-quotes, that the Web server might otherwise think were part of a PHP command or text string.*

Print with double vs. single quotes

1. Create a new script with this code:

```
<?php

# Printing with Double Quotes (")
# vs. Single Quotes (')

$cars_on_lot = 100;

print "<p>Welcome to <b>ACME AUTO!</b></p>\n";

# double quotes

print "<p>Which one of our $cars_on_lot cars is
right for you?</p>\n";

# single quotes

print '<p>Which one of $cars_on_lot is right
for you?</p>\n';

?>
```

2. Save the script as quotes.php in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- `print "<p>Which one of our $cars_on_lot cars is right for you?</p>\n";`

  By using "double quotes" in the above print statement, the number assigned to the variable `$cars_on_lot` (100) is printed to the browser window.

- `print '<p>Which one of $cars_on_lot is right for you?</p>\n';`

  By using ‘single quotes’ in the above print statement, the text **$cars_on_lot** is printed to the browser window along with the words that surround it.

3. Upload **quotes.php** to the **phpscripts** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **quotes.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/quo
tes.php">7. Double vs. single quotes.</a></p>
```

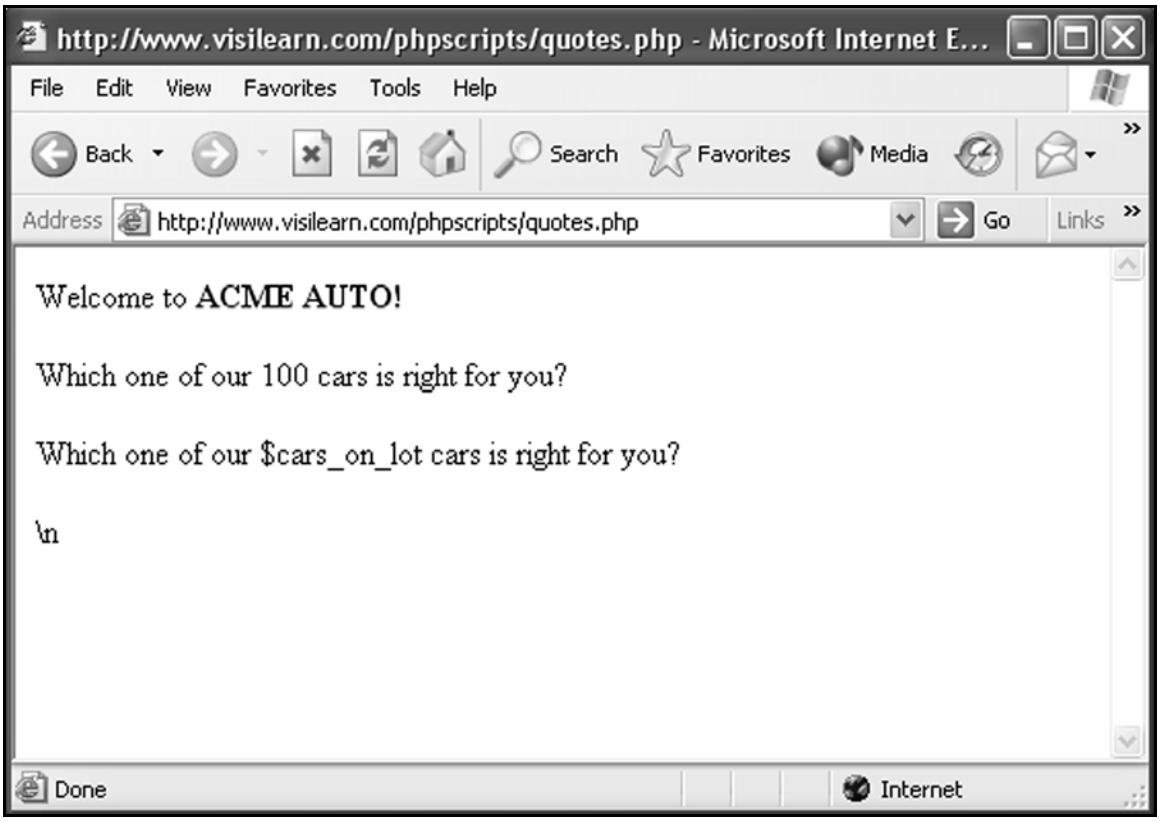
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

www.yourwebsite.com/phplinks.html

7. Click the **7. Double vs. single quotes** link.

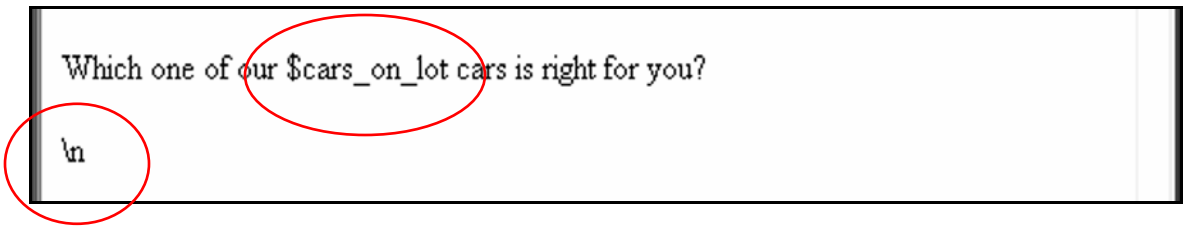
The output should look like this:



Tip: *Using single quotes (`) with the* `print` *function*

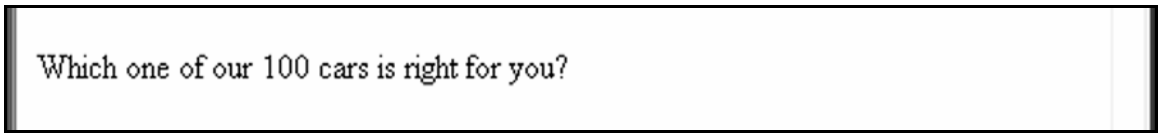
```
print '<p>Which one of $cars_on_lot is right
for you?</p>\n';
```

prints literally everything in between the two quotation marks. (Except HTML tags, such as the *and* *tags.)*



If you want to display the value of a variable, use double quotes (**"**):

```
print "<p>Which one of our $cars_on_lot cars is
right for you?</p>\n";
```



Employ lists of variables

Create lists of number variables

1. Create a new script with this code:

```
<?php

# This script demonstrates how to
# create a numeric array.

$AcmeInventory = array(178,286,387);

print “$AcmeInventory[0]<br>\n”;
print “$AcmeInventory[1]<br>\n”;
print “$AcmeInventory[2]<br>\n”;

print "<p>We just created a list of numbers
using an array variable!";

?>
```

Tip: *Lists of variables are called arrays.*

2. Save the script as **numberlist.php** in the **PHPSCRIPTS** folder.

Here’s what the relevant lines in this script do:

- **`$AcmeInventory = array(178,286,387);`**

  The numbers 178, 286, and 387 are assigned to the array variable **`$AcmeInventory`**.

- `print "$AcmeInventory[0]<br>\n";`

  Prints the first number in the array, the number 178.

- `print "$AcmeInventory[1]<br>\n";`

  Prints the second number in the array, the number 286.

- `print "$AcmeInventory[2]<br>\n";`

  Prints the third number in the array, the number 387.

3. Upload **numberlist.php** to the **phpscripts** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **numberlist.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/num
berlist.php">8. Create a list of
numbers.</a></p>
```

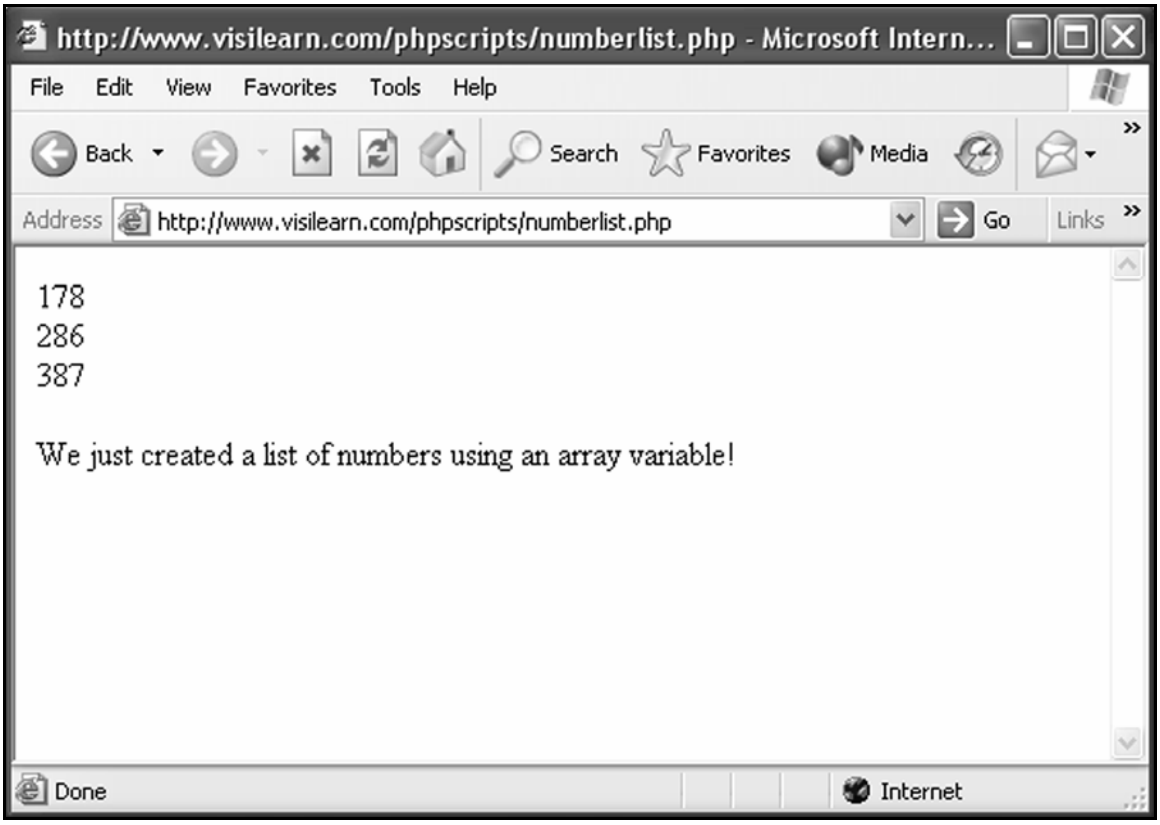
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

   www.yourwebsite.com/phplinks.html

7. Click the **8. Create a list of numbers** link.

The output should look like this:



Create lists of text variables

1. Create a new script with this code:

```
<?php

# This script demonstrates how to
# create a text array.

$AcmeCars = array("Ford","Dodge","Chevy");

print "$AcmeCars[0]<br>\n";
print "$AcmeCars[1]<br>\n";
print "$AcmeCars[2]<br>\n";

print "<p>We have just created a text
array!</p>”;

?>
```

2. Save the script as **textlist.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- **$AcmeCars = ("Ford","Dodge","Chevy");**

  The words "Ford","Dodge", and "Chevy" are assigned to the array variable **`$AcmeCars`**.

- **`print "$AcmeCars[0]<br>\n";`**

  Prints the first word in the array, “Ford.”

- `print "$AcmeCars[1]<br>\n";`

  Prints the second word in the array, “Dodge.”

- `print "$AcmeCars[2]<br>\n";`

  Prints the third word in the array, “Chevy.”

3. Upload **textlist.php** to the **phpscripts** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **textlist.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/tex
tlist.php">9. Create a list of words.</a></p>
```

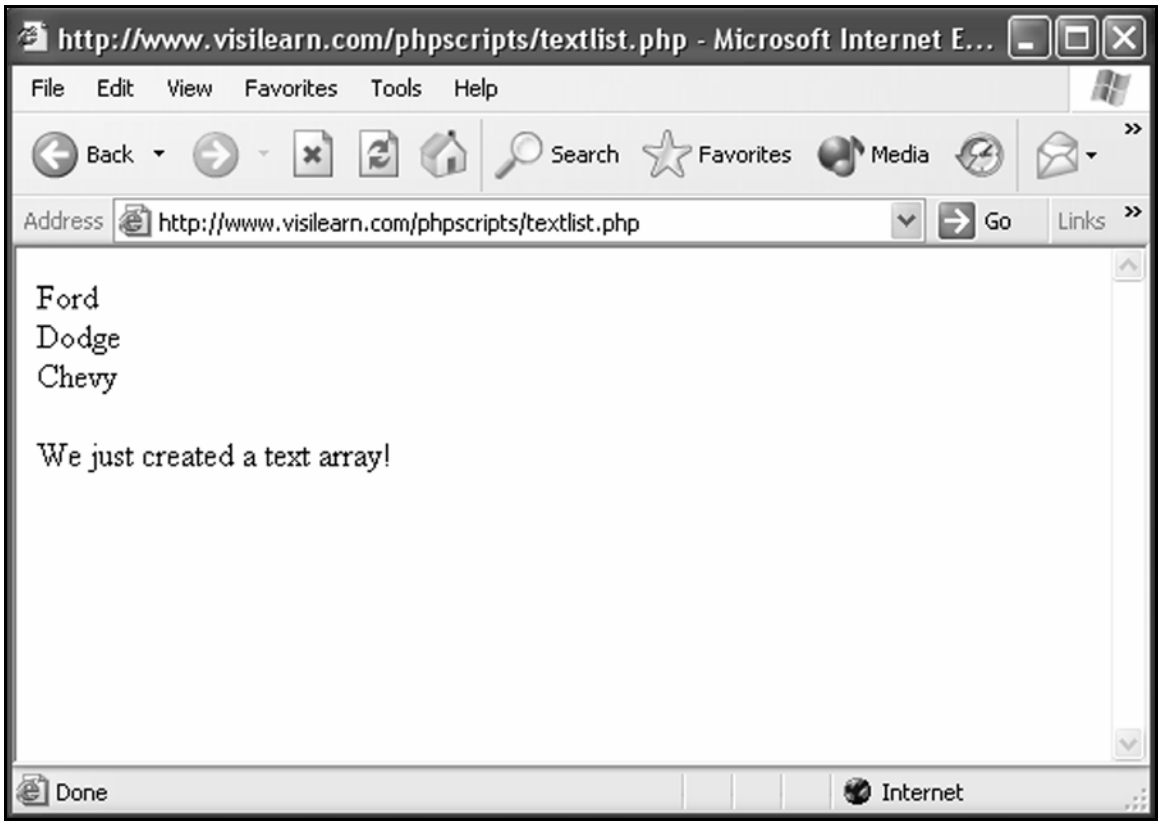
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

   www.yourwebsite.com/phplinks.html

7. Click the **9. Create a list of words** link.

The output should look like this:



Practice:
Working with Variables

1. Write a script that uses a single variable to specify there are 16 monkeys in a barrel of monkeys, then print it to the browser window.

2. Save the script as **monkeys.php** in the **PRACTICEPHP** folder on your computer.

3. Upload it into the **practicephp** directory in your Web site.

4. Add the paragraph "How many monkeys are in a barrel of monkeys?" to **practice.html**, and link that paragraph to **monkeys.php**.

5. Upload **practice.html** to the home directory in your Web site.

6. View practice.html in the browser, then click the new link.

   Its output should look like this:

7. Write a script that creates a list (array) of presidents:

James Buchanan
George Washington
Millard Fillmore

…Prints the text "**This was the Revolution's indispensable man:**"

…Then prints the second name in the list.

8. Save the script as **presidents.php** in the **PRACTICEPHP** folder on your computer.

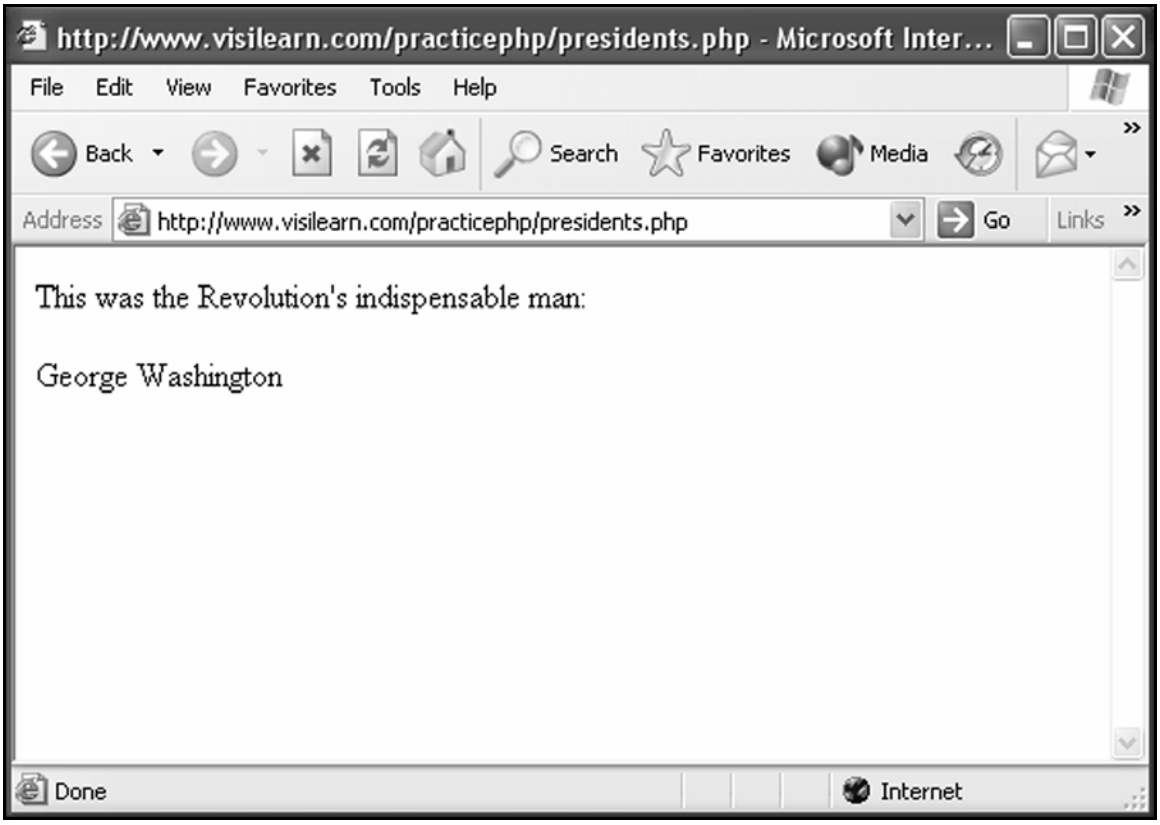
9. Upload it into the **practicephp** directory in your Web site.

10. Add the paragraph "**The indispensable man**" to **practice.html**, and link that paragraph to **presidents.php**.

11. Upload **practice.html** to the home directory in your Web site.

12. View **practice.html** in the browser, then click the new link.

Its output should look like this:



Working with Numbers

In this section, you'll learn how to:

- **Perform calculations**
- **Increment/decrement**
- **Generate random numbers**

Perform calculations

1. Create a new script with this code:

```
<?php

$var1 = 5;
$var2 = 2;

$answer = $var1 + $var2;

print "$var1 plus $var2 equals $answer.";

?>
```

2. Save the script as **add.php** in the **PHPSCRIPTS** folder.

   Here's what the relevant lines in this script do:

   - **`$answer = $var1 + $var2 ;`**

     Adds the variables **`$var1`** and **`$var2`** together, then assigns the sum to a variable called **`$answer`**.

     Since **`$var1`** is 5, and **`$var2`** is 2, **`$answer`** has a value of 7.

3. Upload **add.php** to the **PHPSCRIPTS** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **add.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/add
.php">10. Add five plus two.</a></p>
```

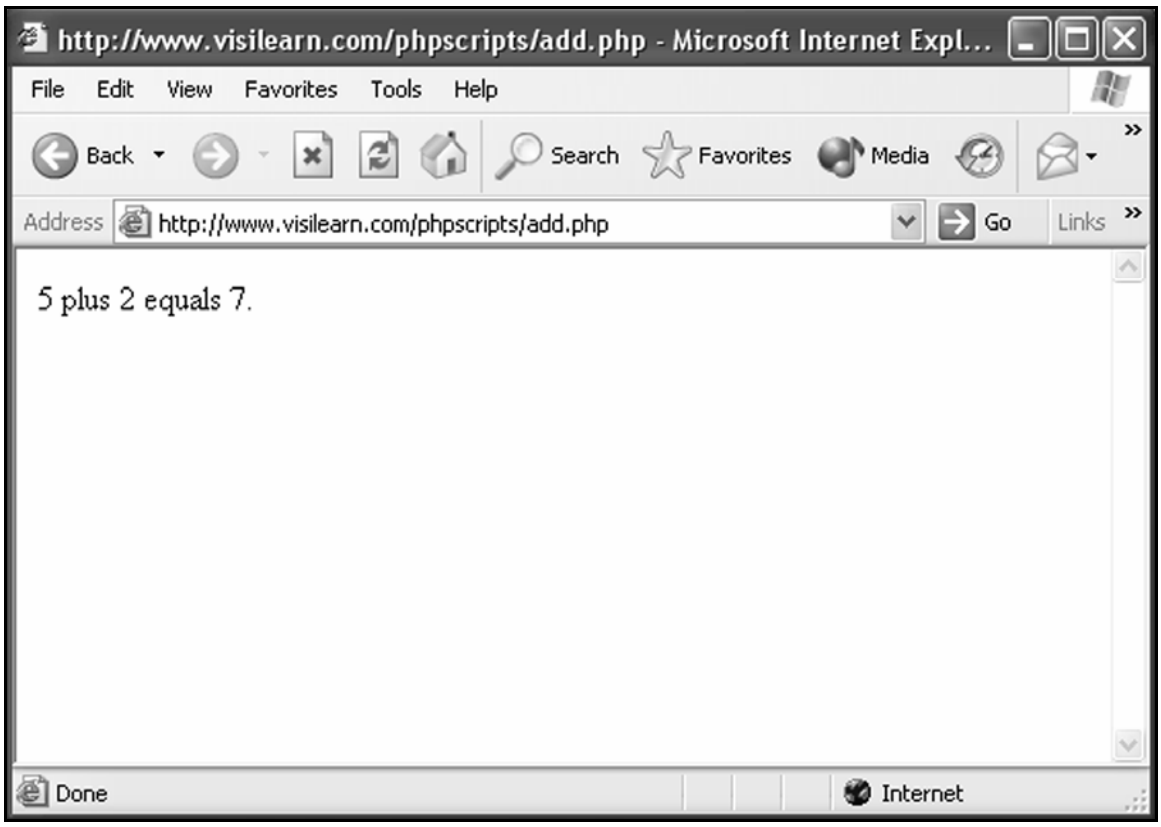
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

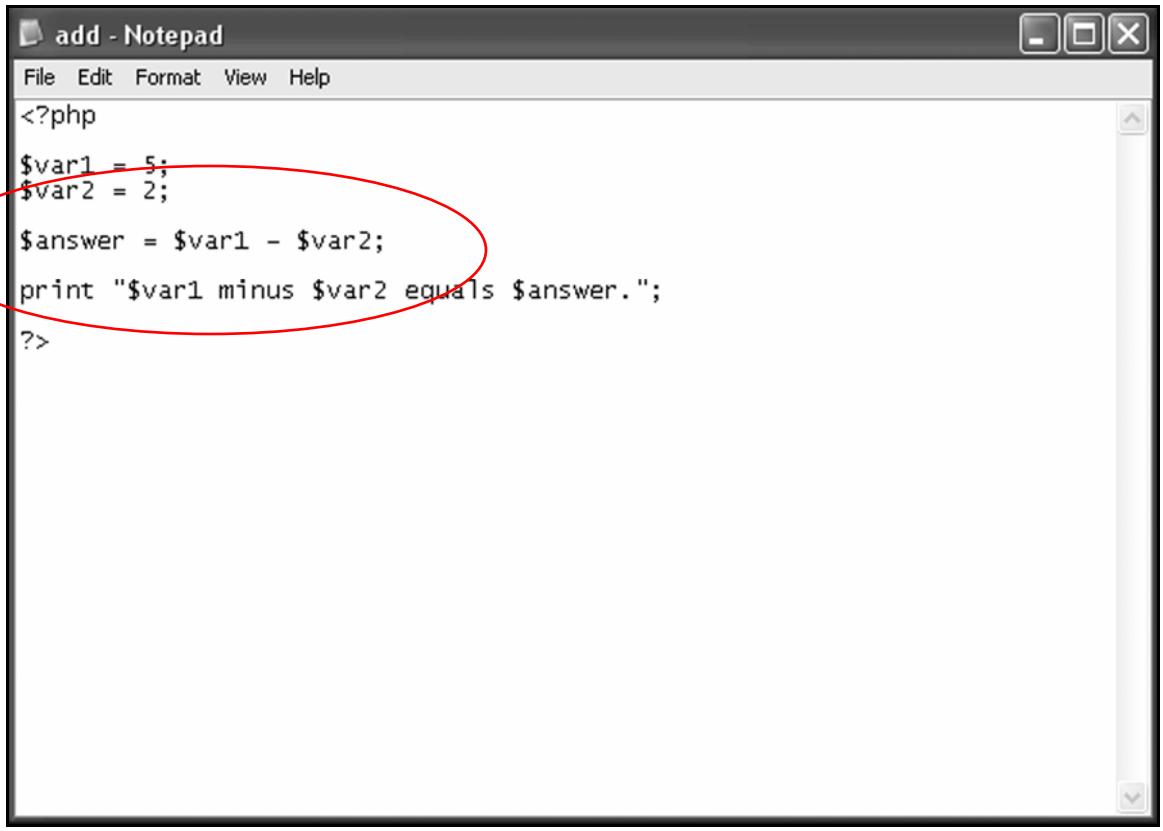
www.yourwebsite.com/phplinks.html

7. Click the **10. Add five plus two** link.

The output should look like this:



Tip: *To subtract, just change the* `+` *sign in the script above to a* `-` *sign.*



To multiply, just change it to a `*` sign.

To divide, change it to `/`.

Increment/decrement

1. Create a new script with this code:

```
<?php

$cars_on_lot = 10;

print "We have $cars_on_lot cars.\n<br>";

print "We got another new car.\n<br>";

$cars_on_lot++;

print "Now we have $cars_on_lot cars!\n<p>";

print '<b>$cars_on_lot++</b> is the same to PHP
as <b>$cars_on_lot + 1.</b>';

?>
```

2. Save the script as **autoplus.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- `$cars_on_lot++;`

  The auto incrementer (`++`) adds 1 to the `$cars_on lot` variable.

- `print '<b>$cars_on_lot++</b> is the same to PHP as <b>$cars_on_lot + 1</b>';`

  Prints the text: **$cars_on_lot++ is the same to PHP as $cars_on_lot + 1.**

3. Upload **autoplus.php** to the **phpscripts** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **autoplus.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/aut
oplus.php">11. Advance a number by 1
automatically.</a></p>
```

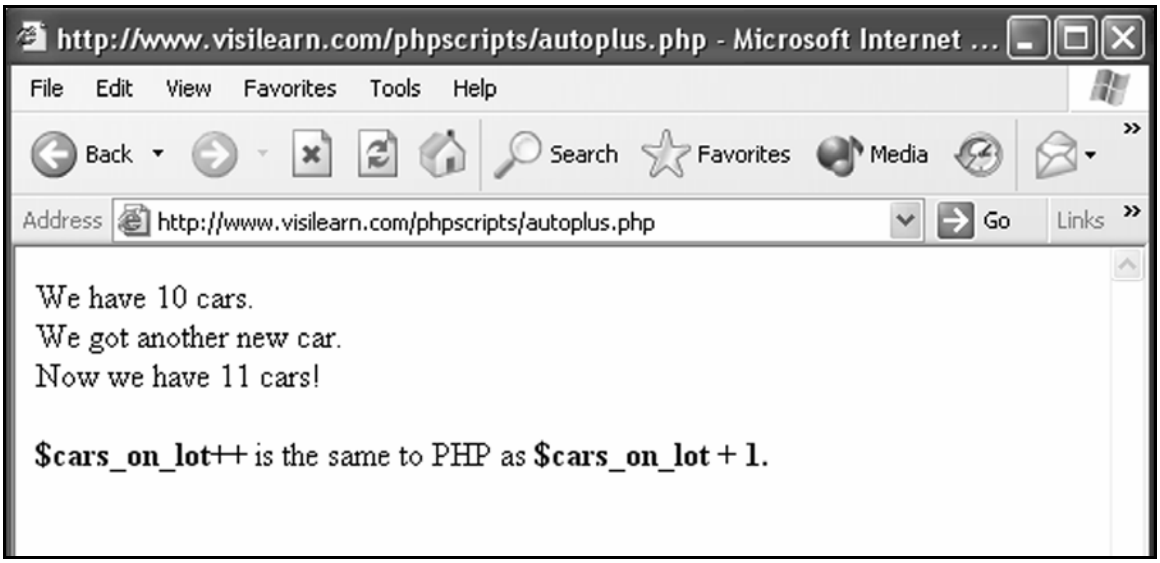
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

   www.yourwebsite.com/phplinks.html

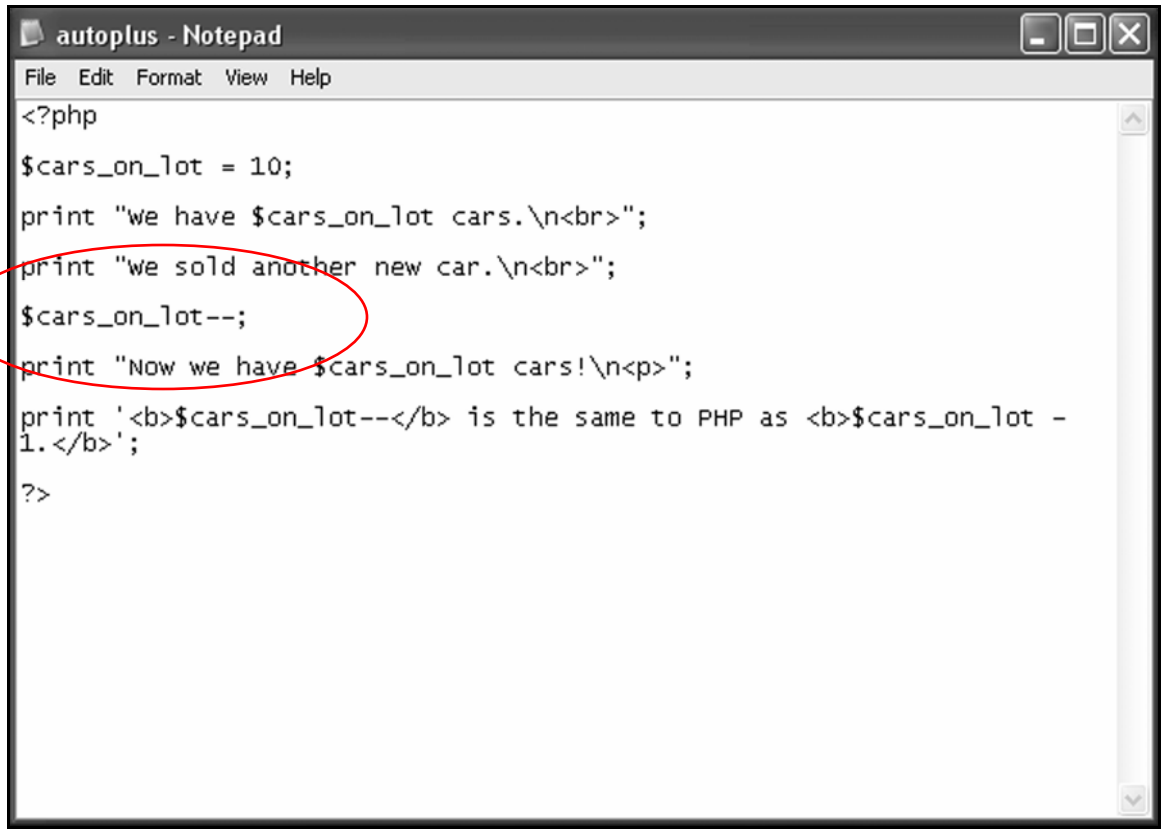
7. Click the **11. Advance a number by 1 automatically** link.

   The output should look like this:



Tip: *To automatically decrement by one, change the auto incrementer in the script above* (++) *to an auto decrementer:*

--



Generate random numbers

1. Create a new script with this code:

```
<?php

$random_number = rand(1,10);

print "<p>Your Lucky Number from 1 to 10 is <b>
$random_number </b></p>";

print "Click the Reload button on your browser
to get a new random number.";

?>
```

2. Save the script as **random.php** in the **PHPSCRIPTS** folder.

   Here's what the relevant lines in this script do:

   - `$random_number = rand(1,10);`

   - `The rand() command generates a random number from 1 to 10.`

     It is then assigned to the variable `$random_number`.

   - `print "<p>Your Acme Auto Lucky Number from 1 to 10 is <b> $random_number </b></p>";`

     Prints "**Your Lucky Number from 1 to 10 is 8**. "

     Because it's a random number, the number on your screen will be different.

3. Upload **random.php** to the **phpscripts** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **random.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/ran
dom.php">12. Generate random numbers.</a></p>
```

5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

   www.yourwebsite.com/phplinks.html

7. Click the **12. Generate random numbers** link.

   The output should look like this:

Practice: Working with Numbers

1. Write a script that multiplies 347 * 221.

2. Save the script as **multiply.php** in the **PRACTICEPHP** folder on your computer.

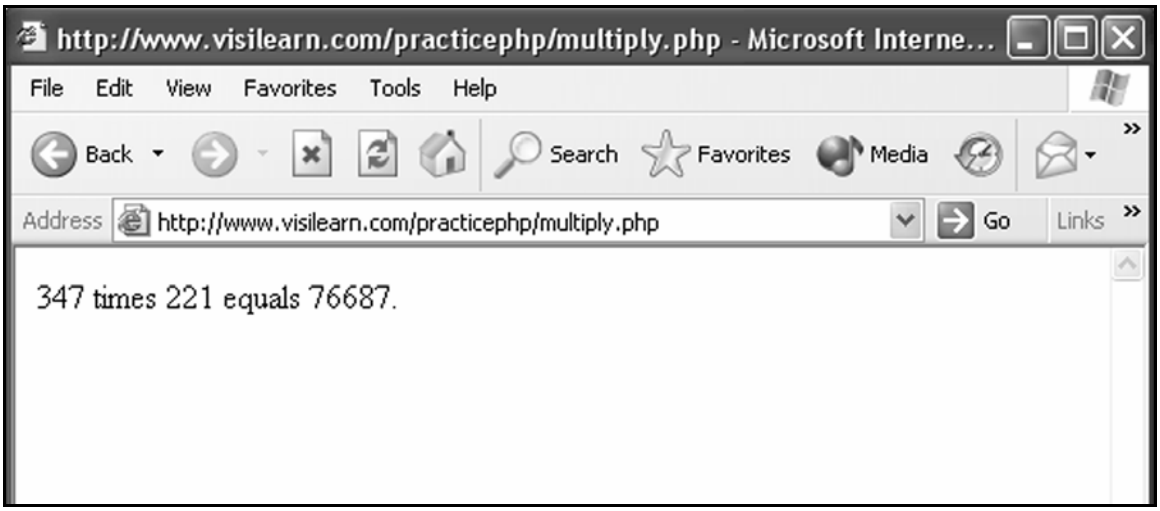
3. Upload it into the **practicephp** directory in your Web site.

4. Add the paragraph "What's 347 times 221?" to **practice.html**, and link that paragraph to **multiply.php**.

5. Upload **practice.html** to the home directory in your Web site.

6. View **practice.html** in the browser, then click the new link.

   Its output should look like this:



7. Write a script that generates a whole random number between 1 and 450.

   Then have the script advance that number by 1.

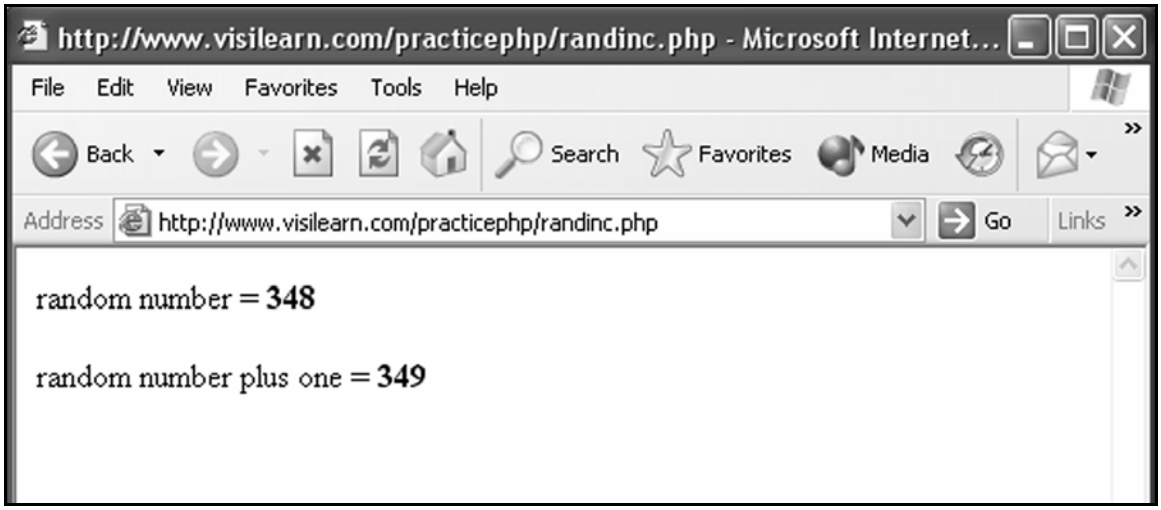
8. Save the script as **randinc.php** in the **PRACTICEPHP** folder on your computer.

9. Upload it into the **practicephp** directory in your Web site.

10. Add the paragraph “random number plus one” to **practice.html**, and link that paragraph to **randinc.php**.

11. View **practice.html** in the browser, then click the new link.

    Its output should look like this:



User Functions

In this section, you'll learn how to:

- **Create a user function**
- **Pass form inputs to a script**

Create a user function

What is a user function?
It's a block of reusable code that you create within your program. It is sometimes called a subroutine.

PHP instructions within the user function can be called or executed from the main program more then once. This makes redundant tasks simpler.

You can also pass variable information into a subroutine to make the user function more powerful and flexible.

1. Create a new page with this code:

```
<html>
<head>
<title>Create a User Function</title>

<?php
  function myFunction($company) {
    print("<p>Welcome to $company</p>");
  }
?>

</head>

<body>

<h3>Start</h3>

<?php
  $company = "Acme Auto";
  myFunction($company);
?>
```

```
<h3>End</h3>

</body>
</html>
```

2. Save the page as **functionsimple.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- **`function myFunction($company) {`**

  `function` is the PHP command to create a function.

  `myFunction` is the name of the user function.

  The variable **`$company`** is passed into the user function inside of the parentheses ( ).

- **`{`**

  Marks the beginning of the user function.

- **`print("<p>Welcome to $company</p>");`**

  This is what the user function does. It prints the text "**Welcome to Acme Auto!**" to the browser window. This is because the variable **`$company`** is given the value of "Acme Auto."

- **`}`**

  Marks the end of the user function.

- **`<body>`**

  This HTML tag begins the part of this script that's viewable in the browser.

- **`<?php`**

  Begins the viewable part of this PHP script.

- **`$company = "Acme Auto";`**

  The variable **`$company`** is assigned the text value "Acme Auto."

- **`myFunction($company);`**

  Executes the code written in the **`myFunction`** user function above. This is where the variable **`$company`** is passed into the user function.

- **`?>`**

  Ends the viewable part of the PHP script.

3. Upload **functionsimple.php** to the **phpscripts** directory in your Web site.

PHP pages

With PHP, you can create HTML pages with PHP scripts embedded in them.

To get them to work, you just give them the extension .php instead of .html.

Example: page.php.

4. Open **phplinks.html** and insert a new link to **functionsimple.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/fun
ctionsimple.php">13. Execute a user
function.</a></p>
```

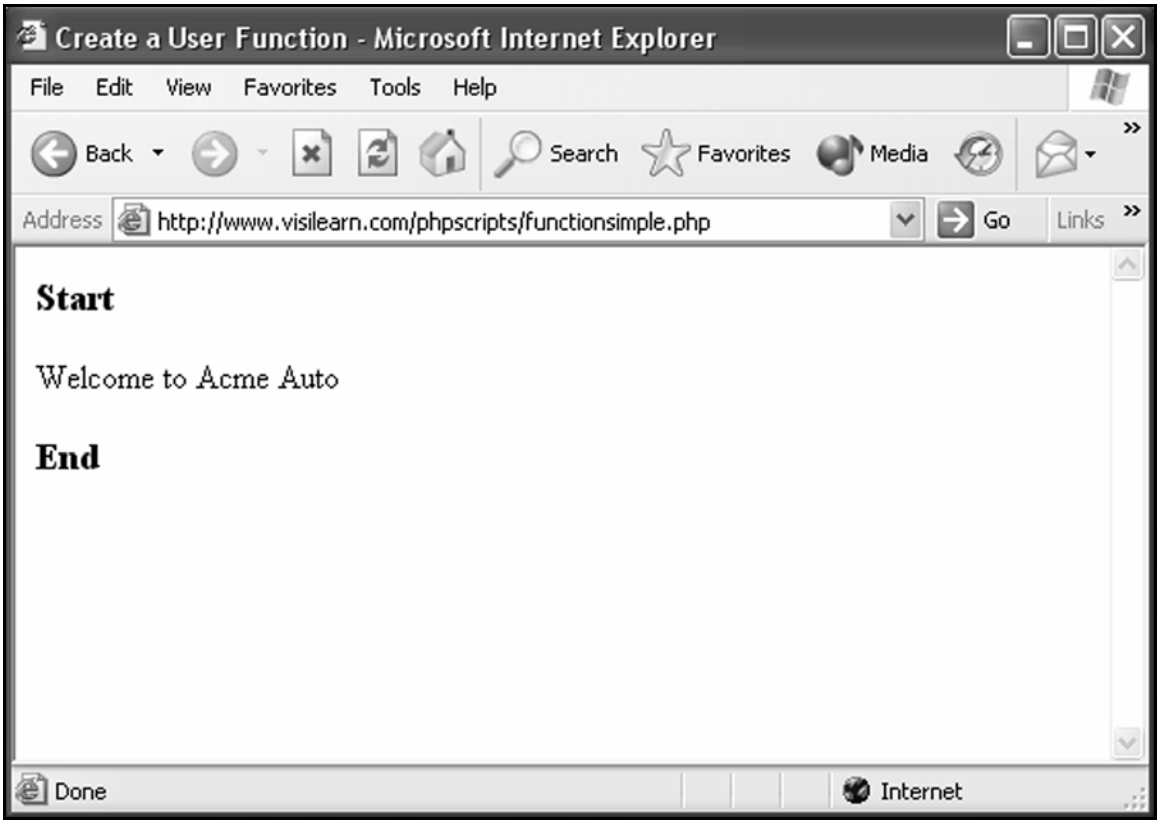
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

www.yourwebsite.com/phplinks.html

7. Click the **13. Execute a user function** link.

The output should look like this:



Pass form inputs to a script

1. Create a new Web page with this code:

```
<html>
<body>

<form method="post" action="
http://www.yourwebsite.com/phpscripts/formoutpu
t.php">

What is your name?

<input type="text" name="yourname"><br>

<input type="submit" value="Submit">

</form>

</body>
</html>
```

2. Save the page as **simpleform.html** in the **PHPSCRIPTS** folder on your computer.

Here's what the relevant lines in this page do:

- `<form method="post" action="
  http://www.yourwebsite.com/phpscripts/for
  moutput.php">`

  When the 

button on the form is clicked, the script **formoutput.php** will execute.

- `<input type="text" name="yourname">`

  This puts a textbox called **`yourname`** on the page.

3. Upload simpleform.html to the home directory in your Web site.

4. Create a new script with this code:

```
<?php

print "Hi $yourname";

?>
```

5. Save the script as **formoutput.php** in the **PHPSCRIPTS** folder on your computer.

6. Here's what the relevant line in this script does:

   - `print "Hi $yourname";`

     Displays the name the user enters on the form page.

     The variable **`$yourname`** is given the value from the textbox called **`yourname`** on the **simpleform.php** page.

7. Upload it into the **phpscripts** directory in your Web site.

8. Open **phplinks.html** and insert a new link to **simpleform.html**:

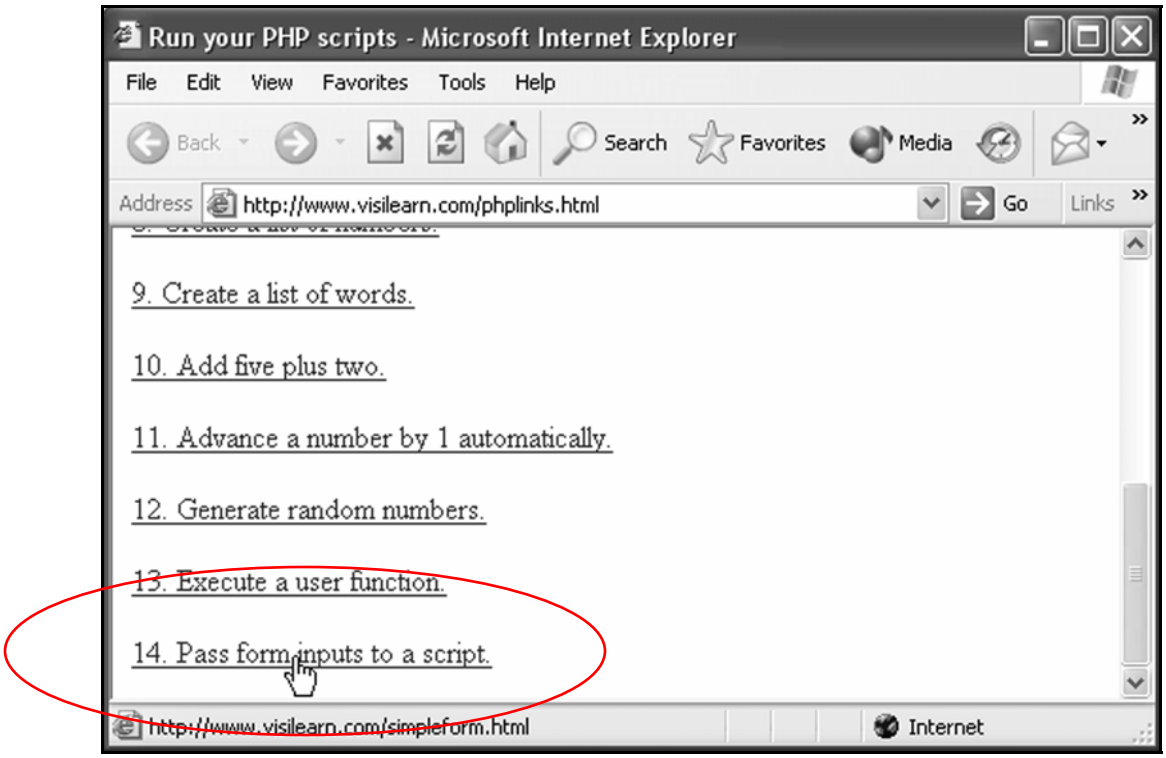
```
<p><a
href="http://www.yourwebsite.com/simpleform.htm
l">14.Pass form inputs to a script.</a></p>
```

9. Save **phplinks.html**, then upload it to the home directory in your Web site.

10. In the browser, go to:

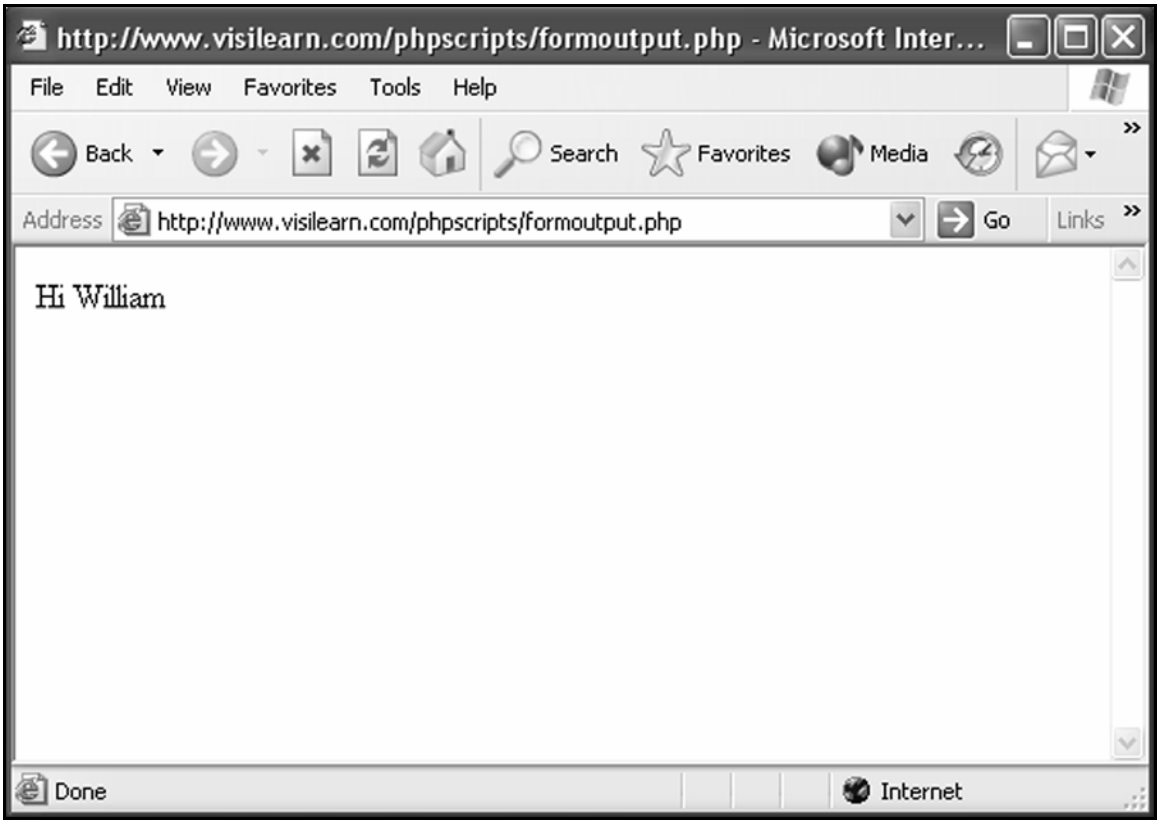
www.yourwebsite.com/phplinks.html

11. Click the **14. Pass form inputs to a script** link.



12. Type your name in the textbox, then click the Submit button.

The output should look like this:



Practice: User Functions

1. Create a copy the **simpleform.html** page, then save the page as **simpleform2.html** in the **PRACTICEPHP** folder on your computer.

2. Modify **simpleform2.html** to call a new php script called **skilltest.php** in the **practicephp** directory, instead of **formoutput.php**.

3. Use the **functionsimple.php** script as a model to create a new script called **skilltest.php**.

4. Make **skilltest.php** print the `$yourname` variable from the **simpleform2.html** page.

5. Upload **simpleform2.html** to the home directory in your Web site.

6. Upload **skilltest.php** to the **practicephp** directory in your Web site.

7. View **simpleform2.html** in the browser, and test the **skilltest.php** script.

Its output should look something like this:

Logic & Loops

In this section, you'll learn how to:

- **Employ conditional logic**
- **Employ looping**

Employ conditional logic

If statements

1. Create a new Web page with this code:

```
<html>
<head>
<title>If Statements</title>
</head>

<body>

<h2>Acme Logon Page</h2>

<form method="POST" action="
http://www.yourwebsite.com/phpscripts/if.php”>

<h2>Enter Password:</h2>

Password: <input type="password" name=
"password"><p>

<input type="submit" value="Submit">

</form>

</body>
</html>
```

2. Save the page as **if.html** in the **PHPSCRIPTS** folder.

3. Upload it to the home directory in your Web site.

4. Create a new script with this code:

```
<?php

$GoodPassword = 'acme';

if ($password == $GoodPassword){
  print "<b>Acme Password verified!</b>\n";
}

?>
```

5. Save the script as **if.php** in the **PHPSCRIPTS** folder, then upload it to the **phpscripts** directory in your Web site.

Here's what the relevant lines in this script do:

- `$GoodPassword = 'acme';`

  Assigns the value “acme” to the variable `$GoodPassword`.

- `if ($password == $GoodPassword){`

  Compares the password word typed in the text box on **if.html** to the password assigned to the variable `$GoodPassword`.

  If they’re the same, then the code between the curly braces is executed:

  ```
  print "<b>Acme Password
  verified!</b><br>\n";
  ```

  Tip: *The double equal sign* `==` *is used to compare variables. A single equal sign assigns a value to a variable.*

6. Open phplinks.html and insert a new link to if.html:

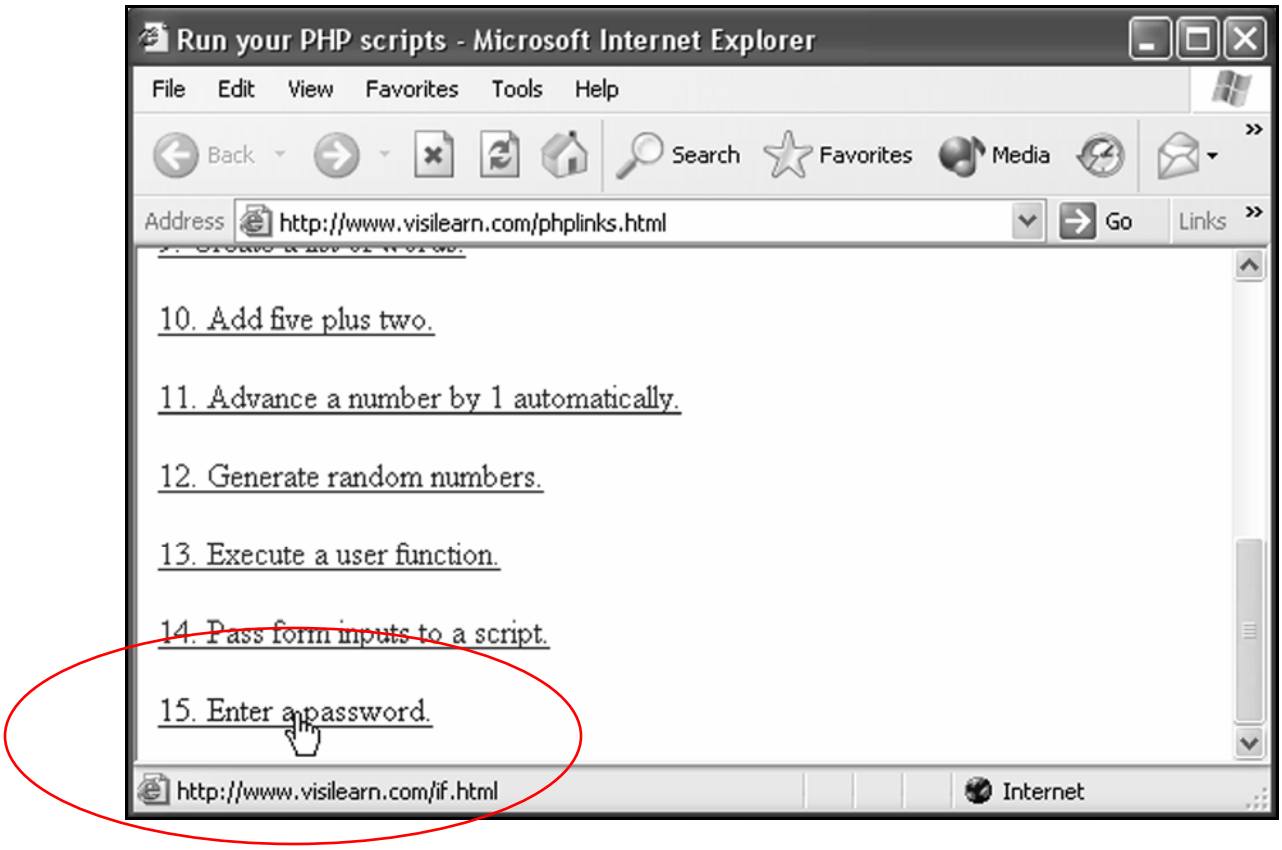
```
<p><a
href="http://www.yourwebsite.com/if.html">15.
Enter a password.</a></p>
```

7. Save **phplinks.html**, then upload it to the home directory in your Web site.

8. In the browser, go to:

www.yourwebsite.com/phplinks.html

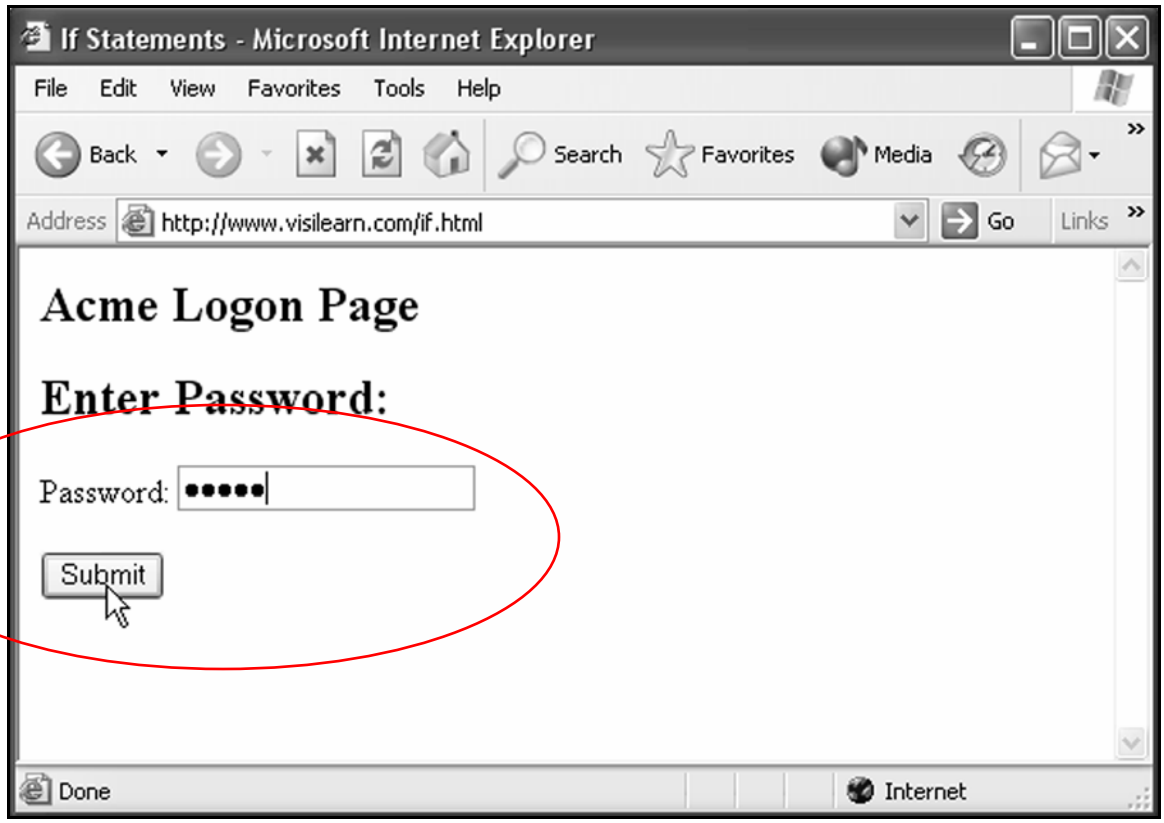
9. Click the **15. Enter a password** link.



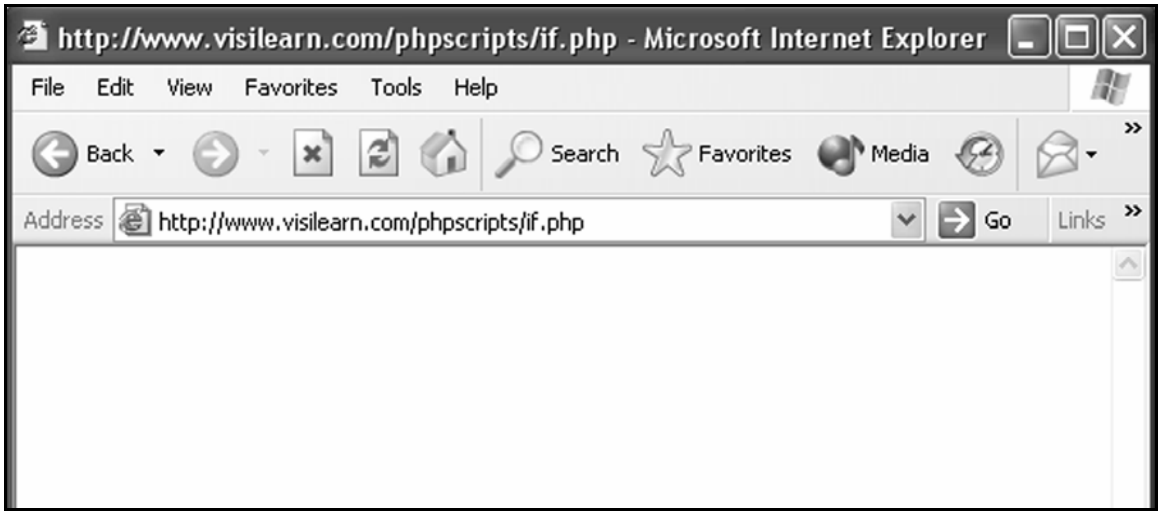
10. In the Password box, type:

pizza

then click the Submit button.



The output should look like this:



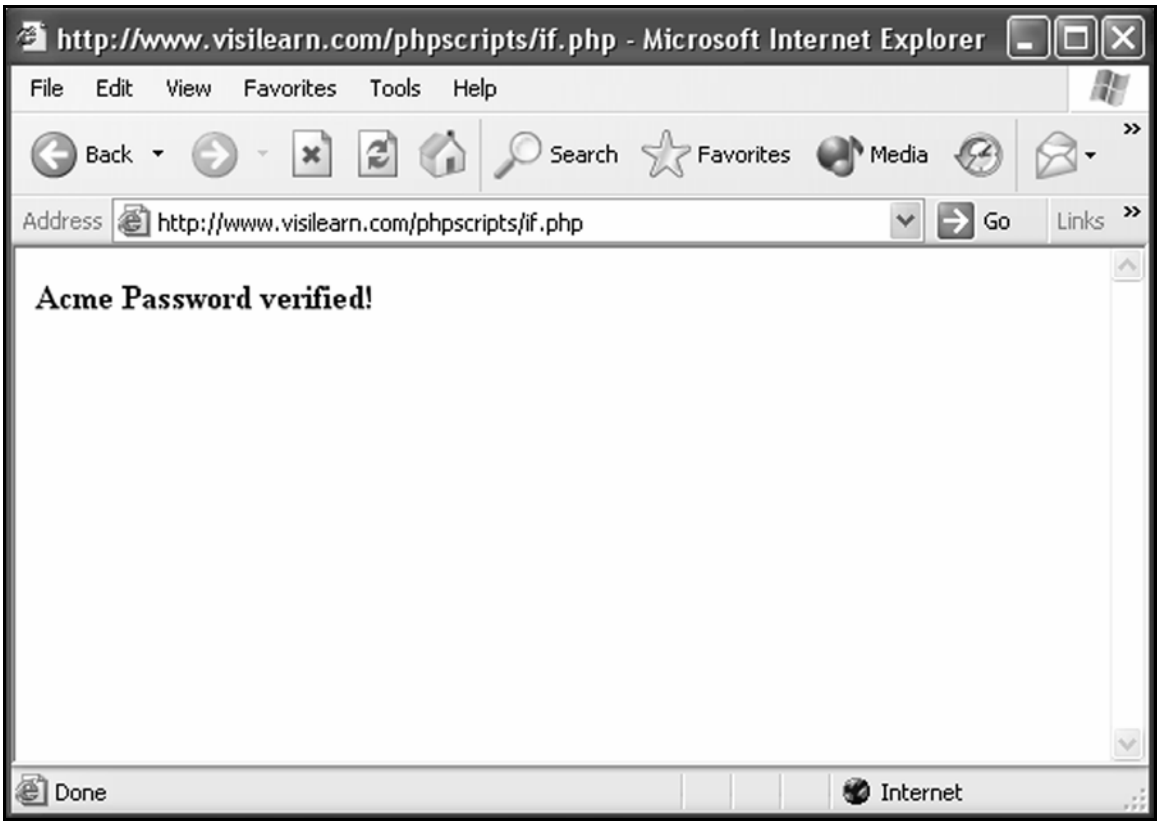
11. Go back to **if.html**.

In the Password box, type:

acme

then click the Submit button.

The output should look like this:



If/else statements

1. In the Web page **if.html**, change the action of its **<form>** tag to use a script called **ifelse.php**:

```
<form method="post"
action="http://www.yourwebsite.com/phpscripts/i
felse.php">
```

2. Save the page as **ifelse.html** in the **PHPSCRIPTS** folder, then upload it to the home directory in your Web site.

3. In the script **if.php**, change its code from this:

```
if ($password == $GoodPassword){
print "<b>Acme Password verified!</b>\n";
}
```

To this:

```
if ($password == $GoodPassword){
print "<b>Acme Password verified!</b>\n";
}
else {
print "<b>Acme Password incorrect.</b>\n";
}
```

4. Save the script as **ifelse.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- `if ($password == $GoodPassword){`

  Compares the password word typed in the text box on **ifelse.html** to the password assigned to the variable `$GoodPassword`.

  If they're the same, then the code between the curly braces is executed.

- `else {`

  If the two values are NOT the same, then the else condition is executed.

  Tip: *Think of it this way:*

  ```
  if (this is true) { then do this }

  else { do this }
  ```

5. Upload **ifelse.php** to the **phpscripts** directory in your Web site.

6. Open **phplinks.html** and insert a new link to **ifelse.html**:

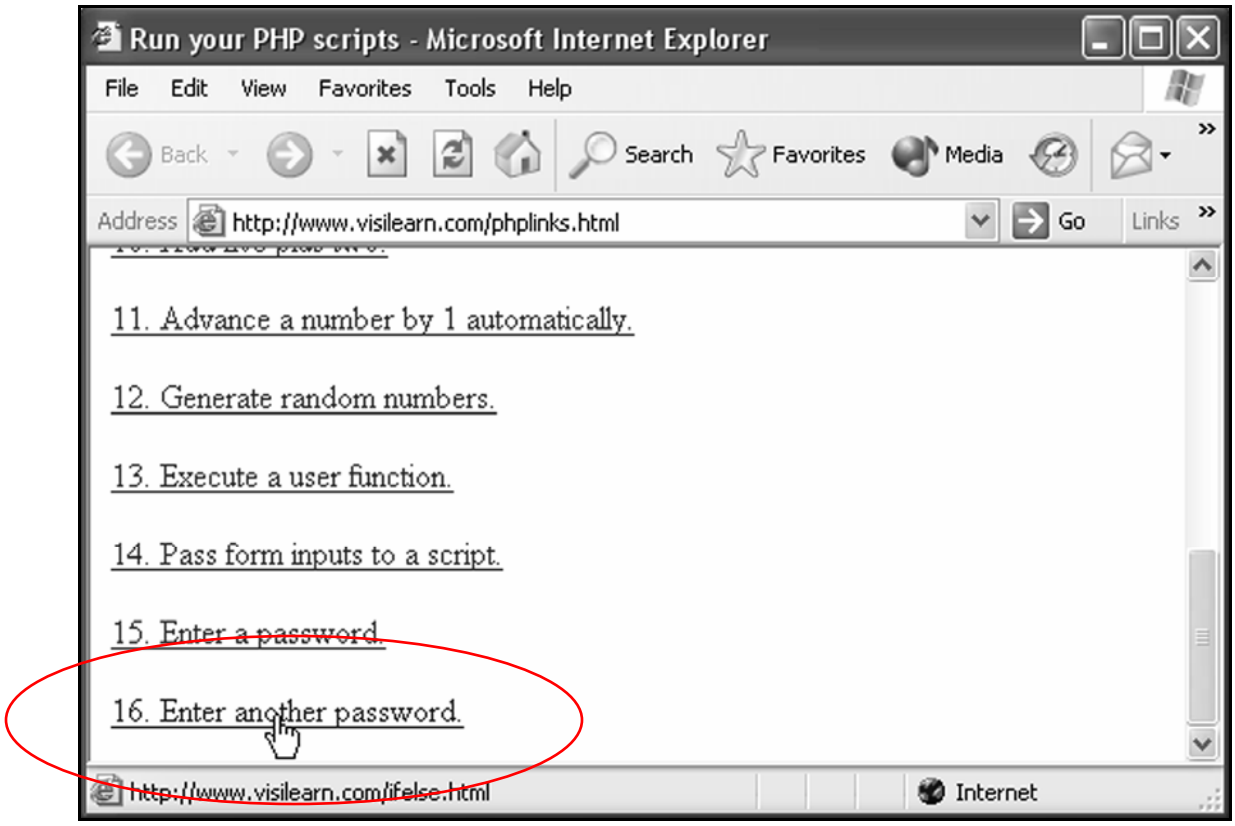
```
<p><a
href="http://www.yourwebsite.com/ifelse.html">1
6.Enter another password.</a></p>
```

7. Save **phplinks.html**, then upload it to the home directory in your Web site.

8. In the browser, go to:

www.yourwebsite.com/phplinks.html

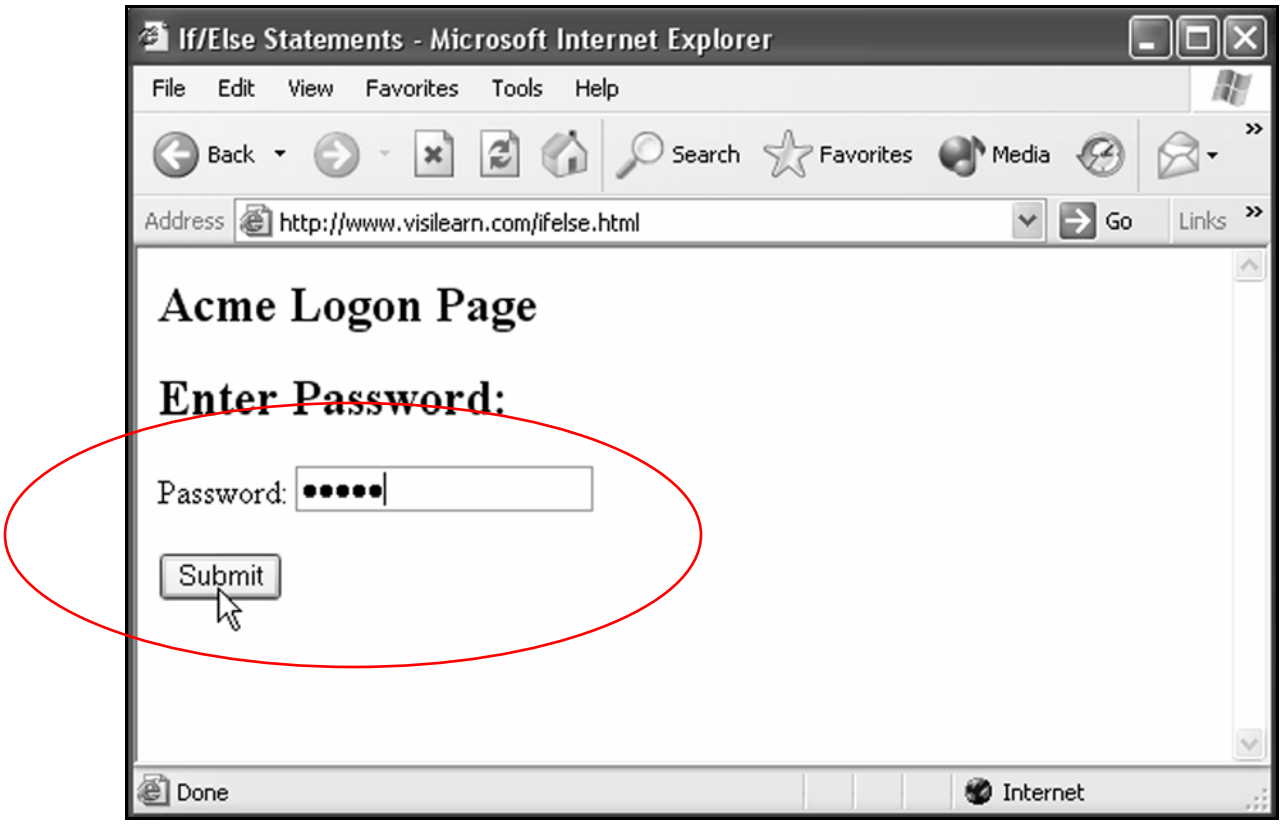
9. Click the **16. Enter another password** link.



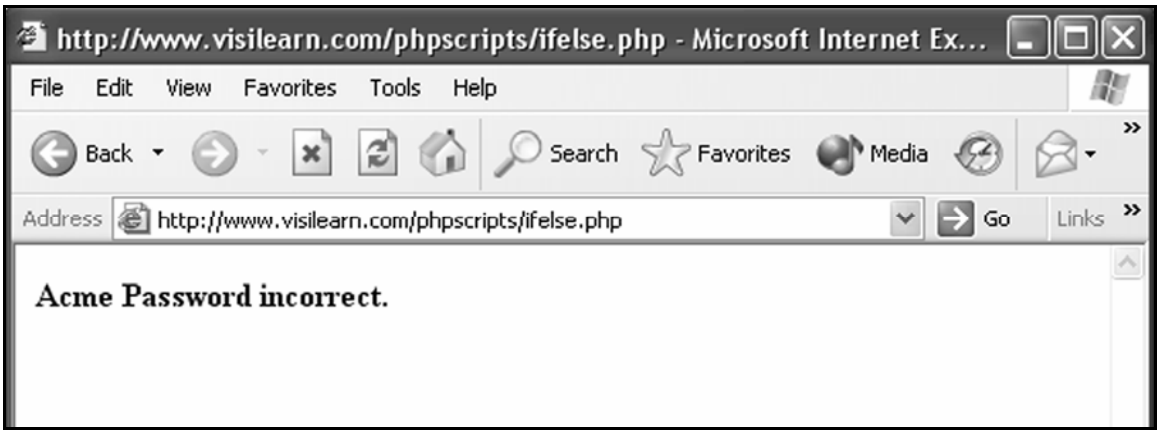
10. In the Password box, type:

pizza

then click the Submit button.



The output should look like this:

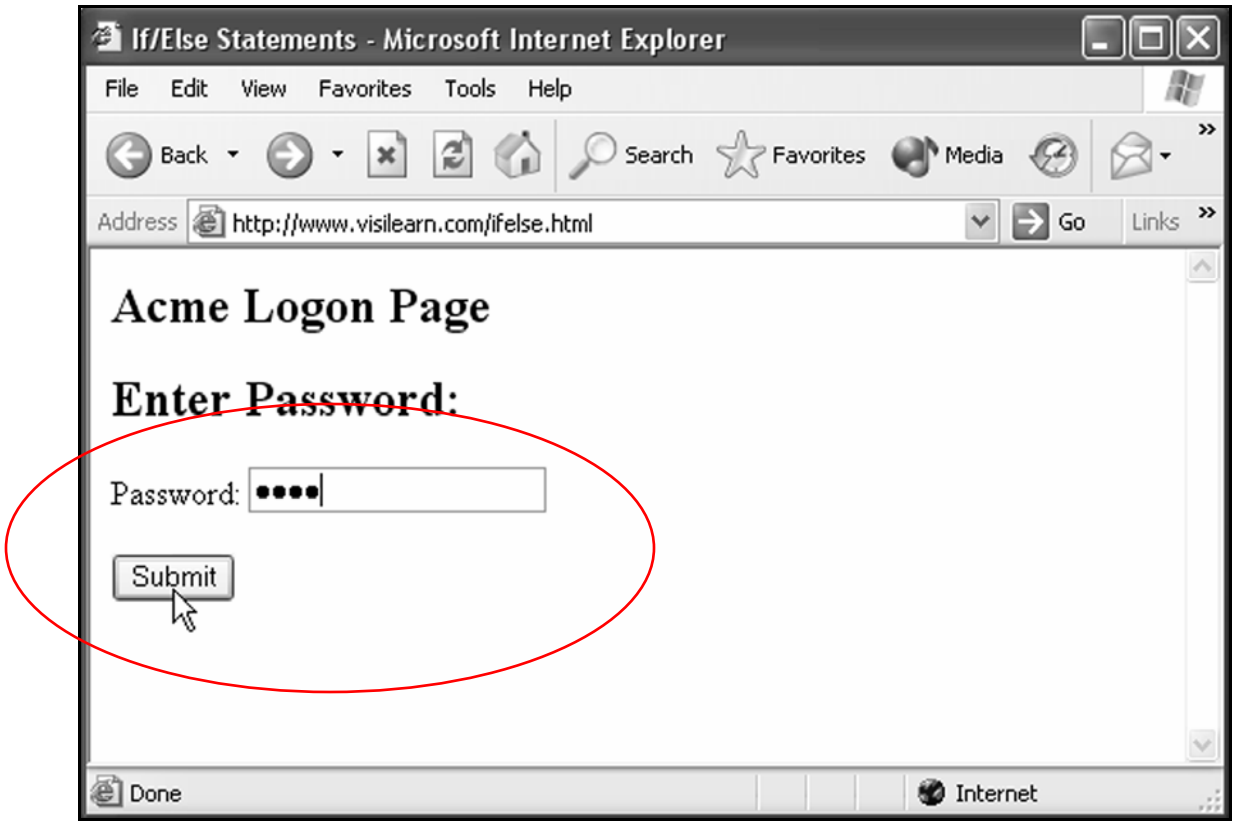


11. Go back to ifelse.html.

In the Password box, type:

acme

then click the Submit button.



The output should look like this:

The OR operator

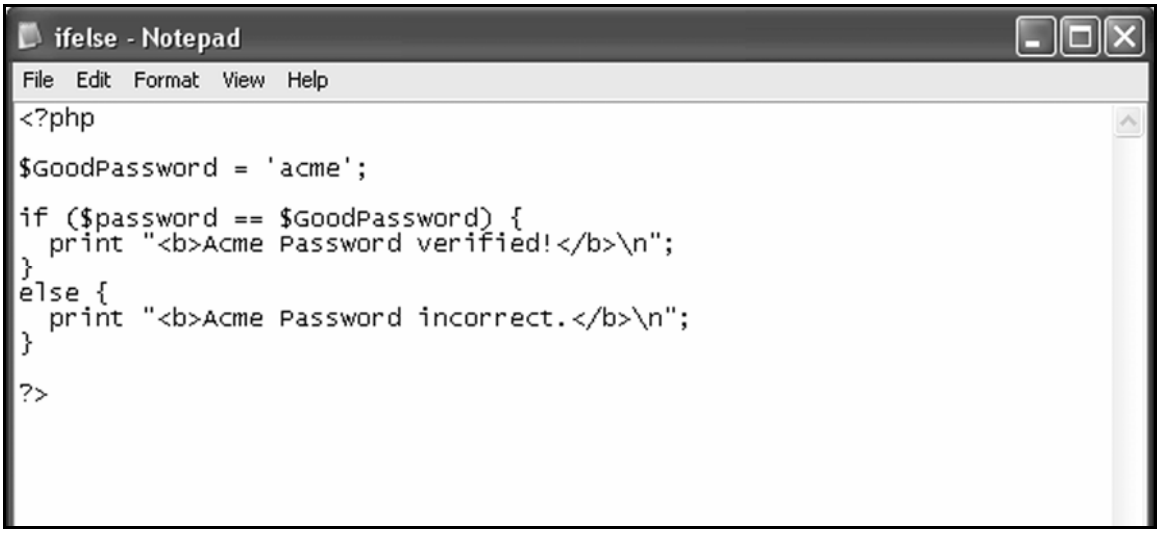
1. Create a new Web page with this code:

```
<html>
<head>
<title>The OR Operator</title>
</head>

<body>

<form method="post" action="
http://www.yourwebsite.com/phpscripts/or.php”>

<h2>Enter Acme Auto User Name</h2>

User Name: <input type="text" name= "username">

<input type="submit" value="Submit">

</form>

</body>
</html>
```

2. Save the page as **or.html** in the **PHPSCRIPTS** folder, then upload it to the home directory in your Web site.

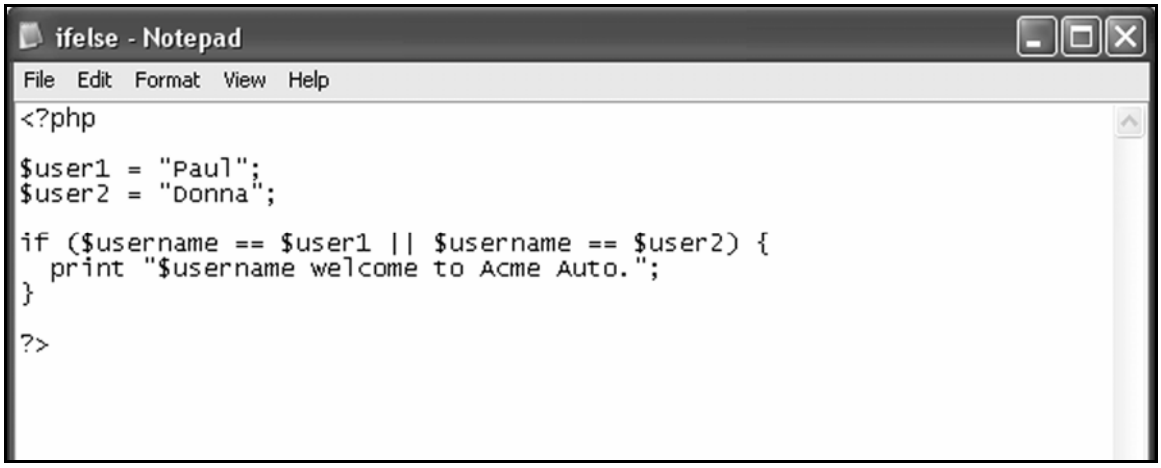
3. Change the code in **ifelse.php** from this:

```
$GoodPassword = 'acme';

if ($password == $GoodPassword){
print "<b>Acme Password verified!</b>\n";
}
else {
print "<b>Acme Password incorrect.</b>\n";
}
```



To this:

```
<?php

$user1 = "Paul";
$user2 = "Donna";

if ($username == $user1 || $username == $user2)
{
print "$username welcome to Acme Auto.";
}

?>
```



4. Save the script as **or.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- **`if ($username == $user1 || $username == $user2)`**

  Uses the OR operator:

  `||`

  to compare two conditions.

  It is asking the question, “is condition one true, OR is condition two true?” Is the entered user name either Paul OR Donna?

  If the entered user name is either Paul or Donna, then the block of code in between the curly braces is executed.

5. Upload **or.php** to the **phpscripts** directory in your Web site.

6. Open **phplinks.html** and insert a new link to **or.html**:

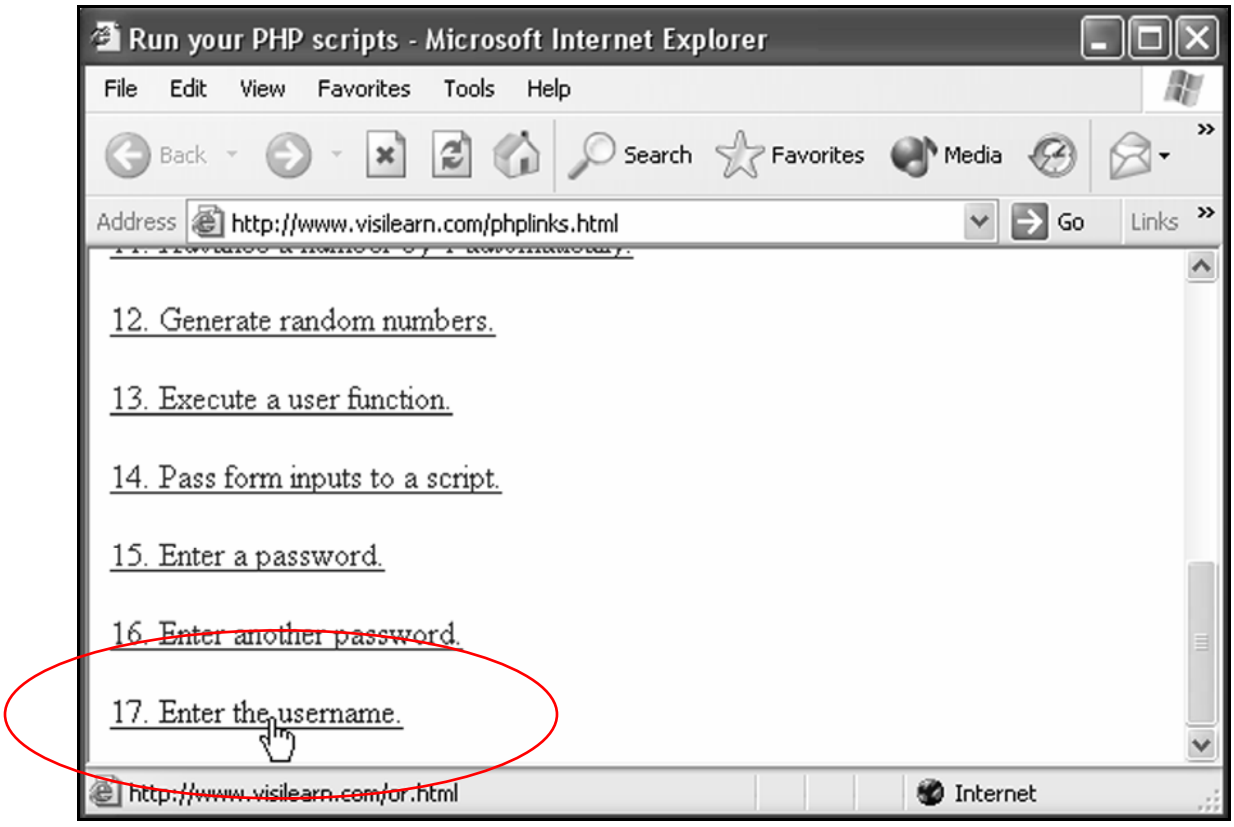
```
<p><a
href="http://www.yourwebsite.com/or.html">17.En
ter the username.</a></p>
```

7. Save **phplinks.html**, then upload it to the home directory in your Web site.

8. In the browser, go to:

www.yourwebsite.com/phplinks.html

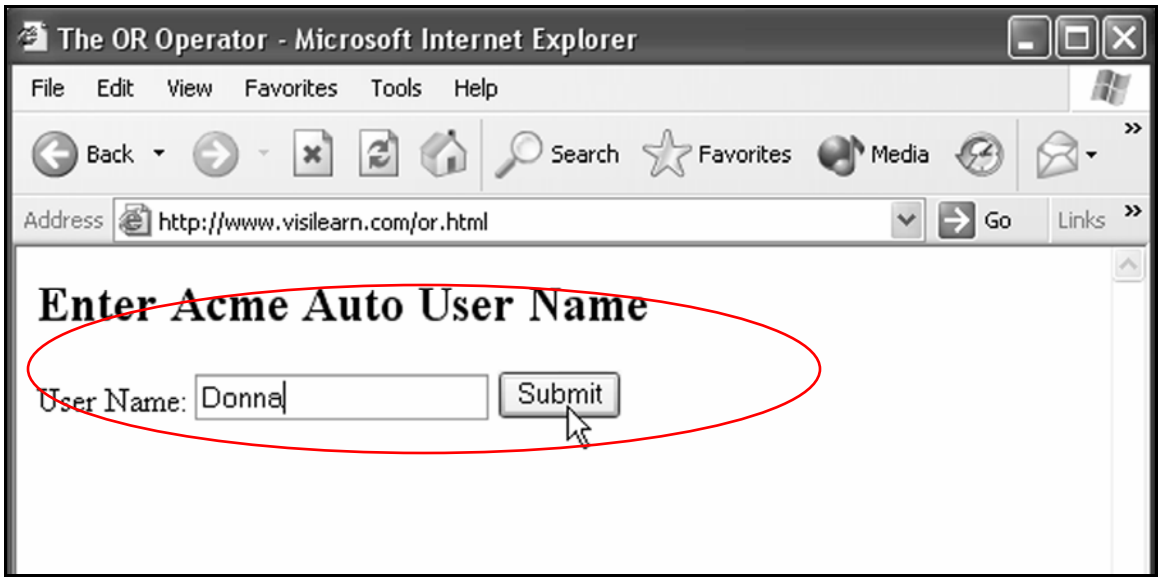
9. Click the **17. Enter the username** link.



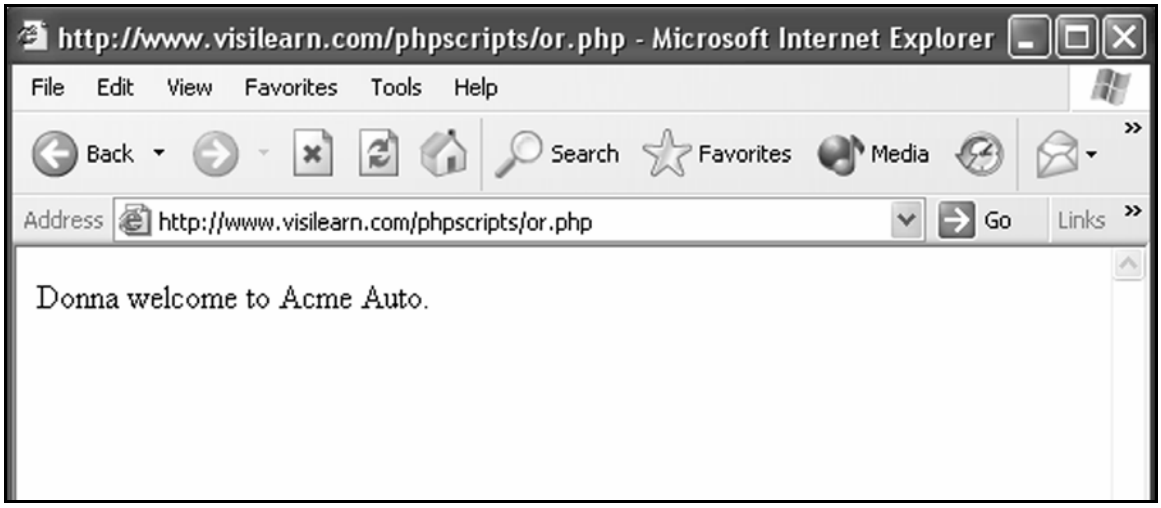
10. In the User Name box, type:

Donna

then click the Submit button.



The output should look like this:

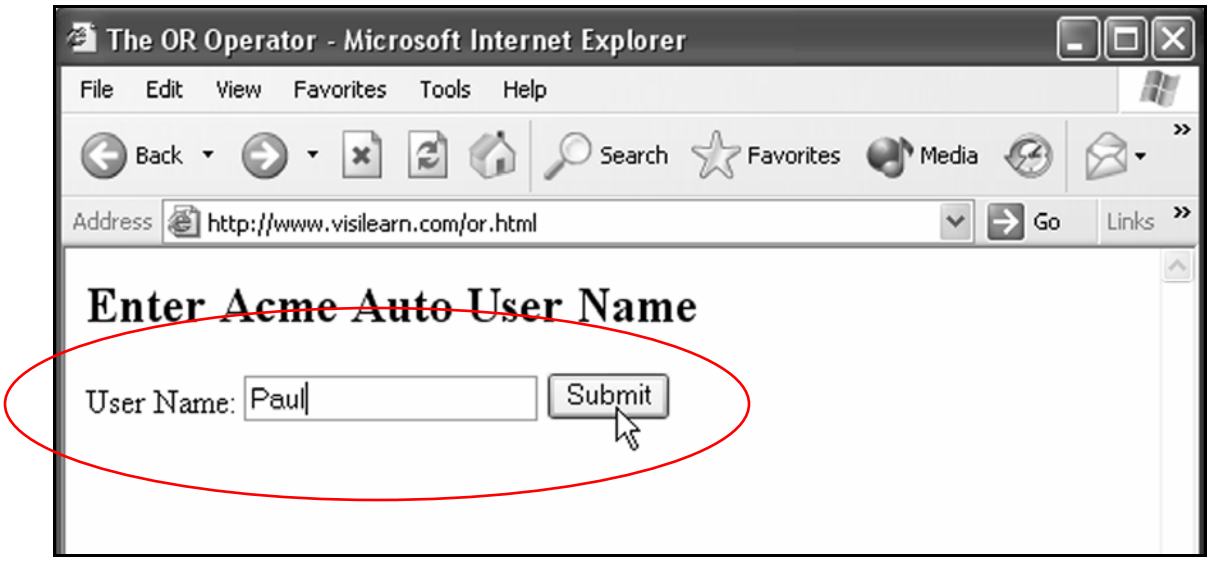


11. Go back to **or.html**.

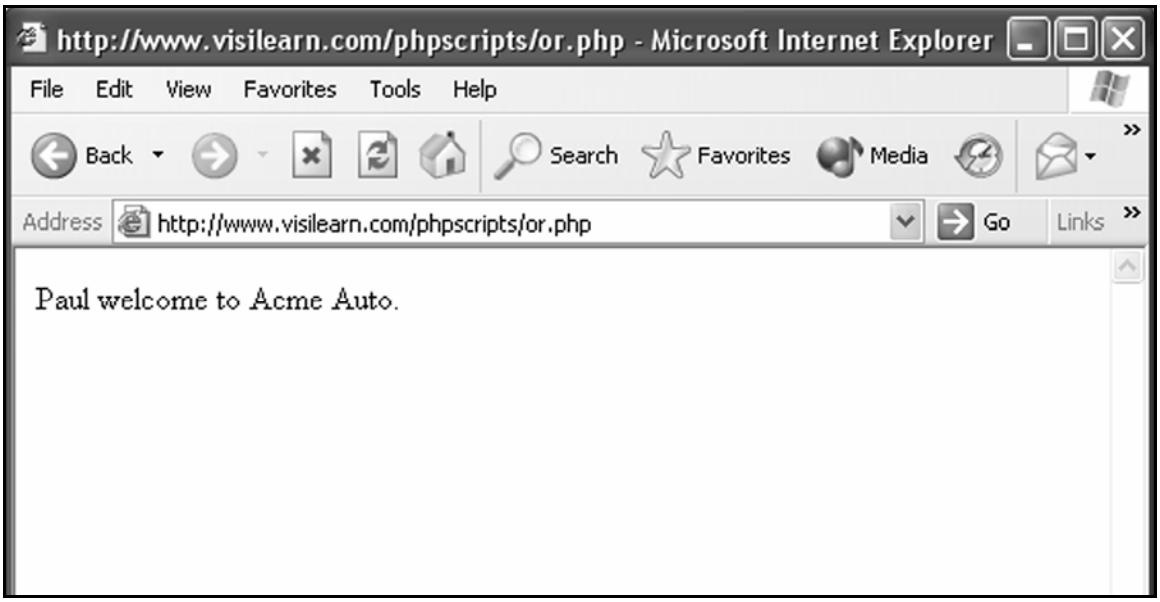
In the User Name box, type:

Paul

then click the Submit button.



The output should look like this:



The AND operator

1. Create a new Web page with this code:

```
<html>
<head>
<title>The AND Operator</title>
</head>
<body>

<form method="post" action="
http://www.yourwebsite.com/phpscripts/and.php">

<h2>Acme Logon Page</h2>

<h3>Enter User Name & Password</h3>

User Name: <input type="text" name=
"username"><br>

Password: <input type="password"
name="password"><br>

<input type="submit" value="Submit">

</form>

</body>
</html>
```

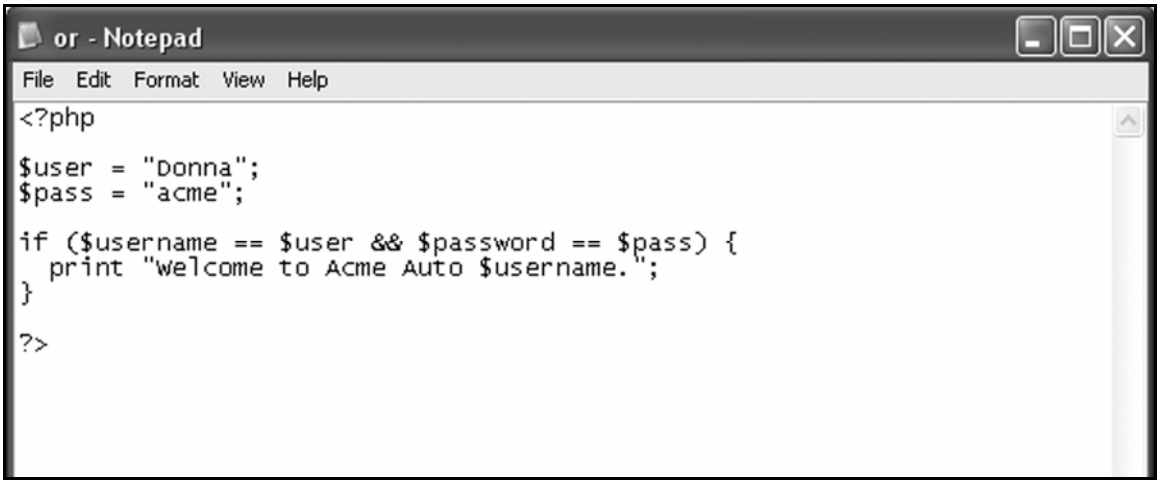
2. Save the page as **and.html** in the **PHPSCRIPTS** folder, then upload it to the home directory in your Web site.

3. Change the code in **or.php** from this:

```
<?php

$user1 = "Paul";
$user2 = "Donna";

if ($username == $user1 || $username == $user2)
{
print "$username welcome to Acme Auto.";
}

?>
```

To this:

```
<?php

$user = "Donna";
$pass = "acme";

if ($username == $user && $password == $pass) {
print "Welcome to Acme Auto $username.";
}

?>
```



4. Save the script as **and.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- **`if ($username == $user && $pass == $pass)`**

  Uses the AND operator:

  `&&`

  to compare two conditions.

  It's asking:

  Is the word entered in the textbox named username the same as the word assigned to the variable **`$user`**?

  AND

  Is the word entered in the textbox named **`password`** the same as the word assigned to the variable **`$pass`**?

  If both these things are true, then execute the code in the curly braces.

5. Upload **and.php** to the **phpscripts** directory in your Web site.

6. Open **phplinks.html** and insert a new link to **and.html**:

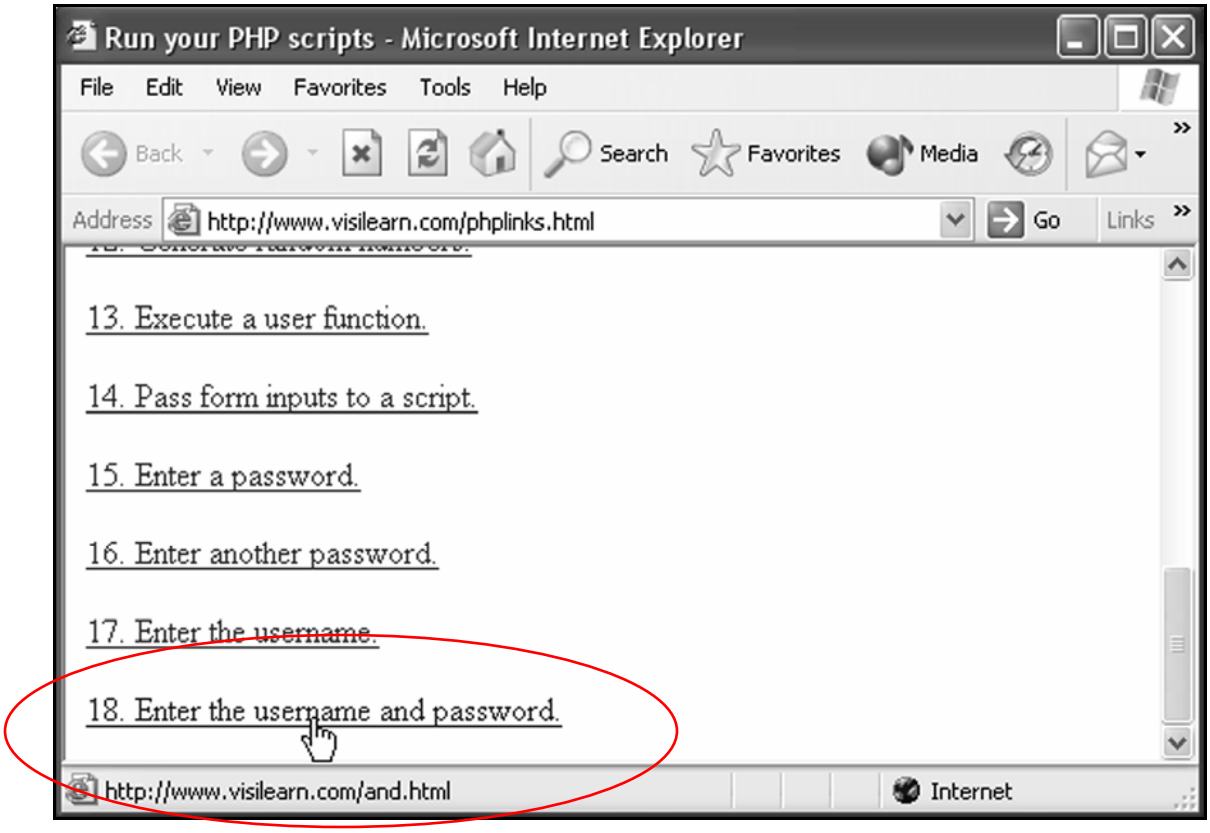
```
<p><a
href="http://www.yourwebsite.com/and.html">18.E
nter the username and password.</a></p>
```

7. Save **phplinks.html**, then upload it to the home directory in your Web site.

8. In the browser, go to:

www.yourwebsite.com/phplinks.html

9. Click the **18. Enter the username and password** link.



10. In the User Name box, type:

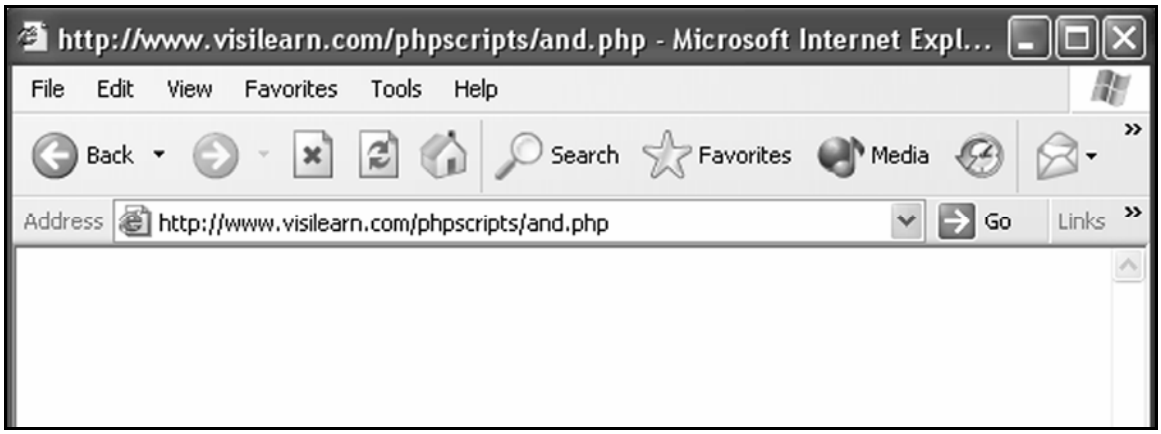
Donna

11. In the Password box, type:

asdf

then click the Submit button.

The output should look like this:



12. View **and.html** in the browser.

Enter **Donna** as the User Name and **acme** as the Password, then click the Submit button.

The output should look like this:

Employ looping

Print a list of elements

1. Create a new script with this code:

```
<?php

$AcmeCars = array("Ford","Dodge","Chevy");

print "<p>The text array contains:</p>";

foreach ($AcmeCars as $thisCar){
    print "$thisCar<br>\n";
}

?>
```

2. Save the script as **printlist.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in the script do:

- **`$AcmeCars = array ("Ford","Dodge","Chevy");`**

  Creates the array variable **`@AcmeCars`** and places the “Ford”, “Dodge”, and “Chevy” values into the array.

- `foreach ($AcmeCars as $thisCar){`

  `foreach` tells the Web server to "loop" through the `$AcmeCars` array, going through each value in the array, one by one.

  `$thisCar` is a variable: it tells the Web server to pull out each separate element in the `$AcmeCars` array.

3. Upload **printlist.php** to the **phpscripts** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **printlist.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/pri
ntlist.php">19. Print a list of
elements.</a></p>
```

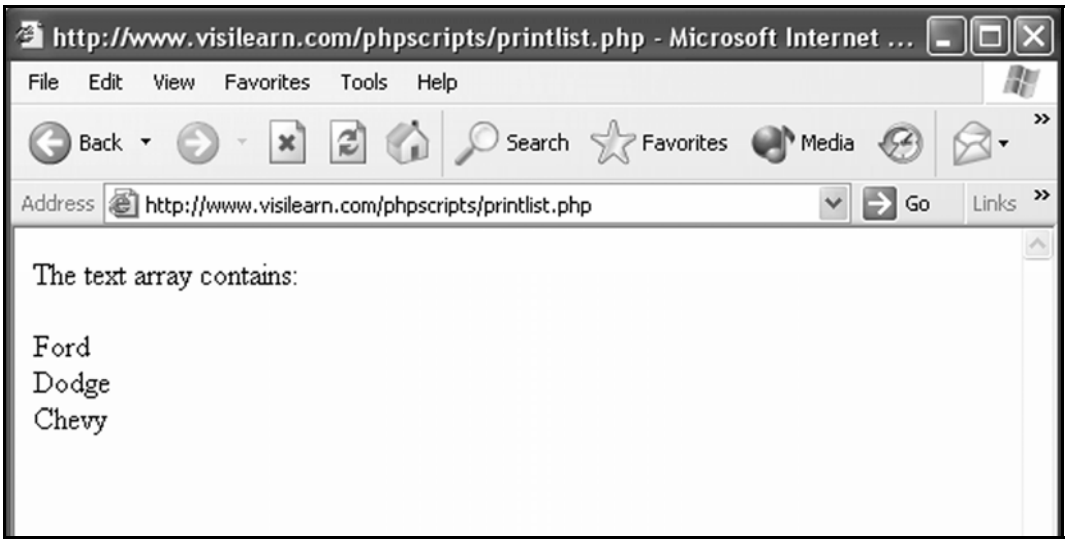
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

www.yourwebsite.com/phplinks.html

7. Click the **19. Print a list of elements** link.

The output should look like this:



Print elements in a table

1. Create a new script with this code:

```
<html>
<head>
<title>Table Example</title>
</head>
<body>
<table border=1 bgcolor=yellow>

<?php

$AcmeCars = array ("Ford","Dodge","Chevy");
$AcmeCarId = array (“K234”,”J987”,”F982”);

for($x=0; $x < count($AcmeCars); $x++){
    print “<tr><td>$AcmeCarId[$x]</td>”;
print “<td>$AcmeCars[$x]</td></tr>\n";
}

?>

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

2. Save the script as **tablelist.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- **`<table border=1 bgcolor=yellow>`**

  HTML tag that sets up an HTML table.

- **`<?php`**

  PHP start tag.

- **`$AcmeCars = array ("Ford","Dodge","Chevy");`**

  Creates the array list called **`$AcmeCars`**. This is a list of cars.

- **`$AcmeCarId = array ("K234","J987","F982");`**

  Creates the array list called **`$AcmeCarId`**. This is a list of ID numbers for each of the cars.

- **`for($x=0; $x < count($AcmeCars); $x++){`**

  This creates a loop. The variable **`$x`** starts at 0 and increases by 1 (**`$x++`**) each time it loops.

  The loop will stop when **`$x`** reaches the **`count()`** of the number of items in the **`$AcmeCars`** variable.

- **`print "<tr><td>$AcmeCarId[$x]</td>";`**

  Prints each **`$AcmeCarId`** item within a table cell each time the program loops around.

- `print "<td>$AcmeCars[$x]</td></tr>\n";`

  Prints each `$AcmeCars` item within a table cell each time the program loops around.

- `}`

  This is the closing `}` for the for loop.

- `?>`

  PHP end tag.

3. Upload **tablelist.php** to the **phpscripts** directory in your Web site.

4. Open **phplinks.html** and insert a new link to **tablelist.php**:

```
<p><a
href="http://www.yourwebsite.com/phpscripts/tab
lelist.php">20. Print a list in a
table.</a></p>
```

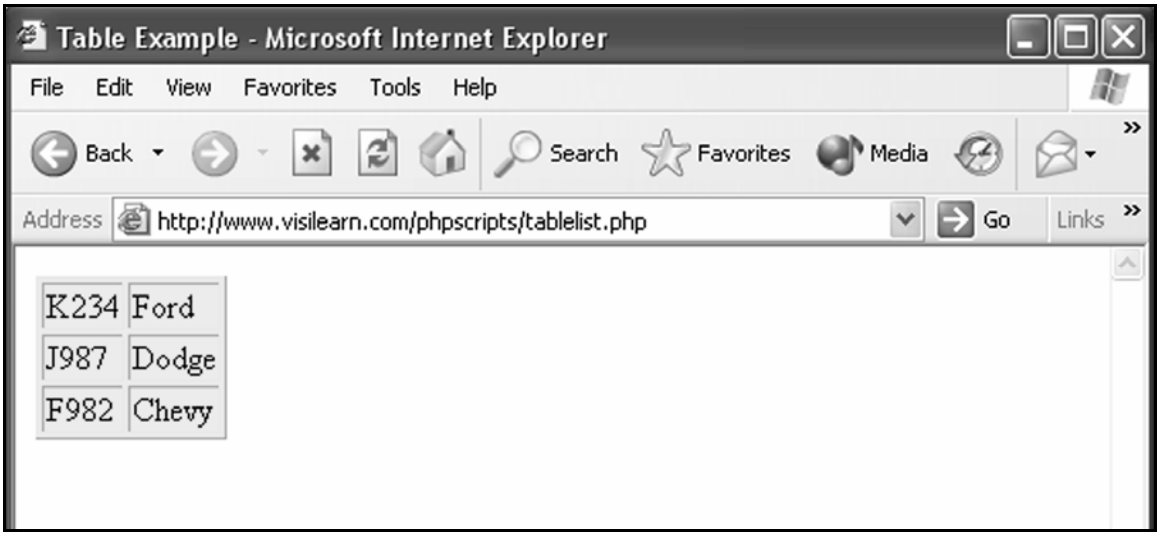
5. Save **phplinks.html**, then upload it to the home directory in your Web site.

6. In the browser, go to:

   www.yourwebsite.com/phplinks.html

7. Click the **20. Print a list in a table** link.

The output should look like this:



Practice: Logic & Loops

1. Create a Web page and script, **entry.html** and **entry.php**, that work together to password-protect a Web page.

Write the script so that IF the proper username/password combination is entered, the user is taken to a page at:

www.yourwebsite.com/presidents.html

Tip: *Refer to the script used in the If/Else statements task, but instead of printing text if the* `if` *condition is fulfilled, have it do this:*

```
print "<script>window.location.replace
(\"http://www.yourwebsite.com/presidents.html\"
)</script>";
```

ELSE, have **entry.php** print this:

Sorry, it didn't work. Try again.

Link the words **Try again** to the page at:

www.yourwebsite.com/entry.html

2. Save **entry.html** and **entry.php** in the **PRACTICEPHP** folder on your computer.

3. Upload **entry.html** to the home directory in your Web site, then upload **entry.php** to the **practicephp** directory in your Web site.

4. Create a page called **presidents.html**, save it in the **PRACTICEPHP** folder, then upload it to the home directory in your Web site.

The page should have this code:

```
<h3>Who were our three greatest
presidents?</h3>

<input type="text" name="prez1">
<p>
<input type="text" name="prez2">
<p>
<input type="text" name="prez3">
```

5. Add a `<form>` tag that specifies the form action as:

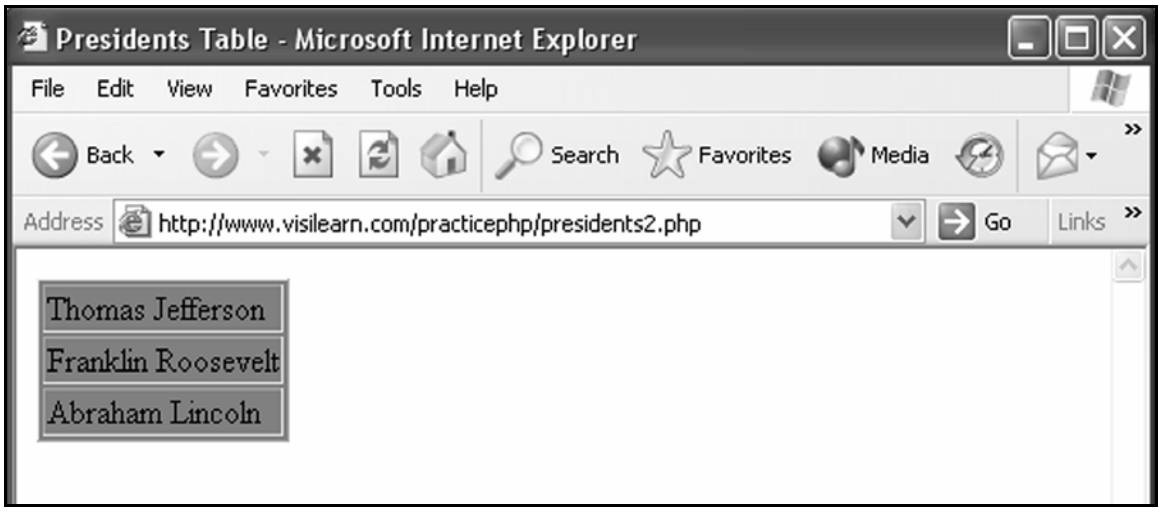
```
http://www.yourwebsite.com/phpscripts/president
s.php
```

6. Create a script called **presidents.php** that:

-Requests the names entered in the input boxes on **presidents.html**.

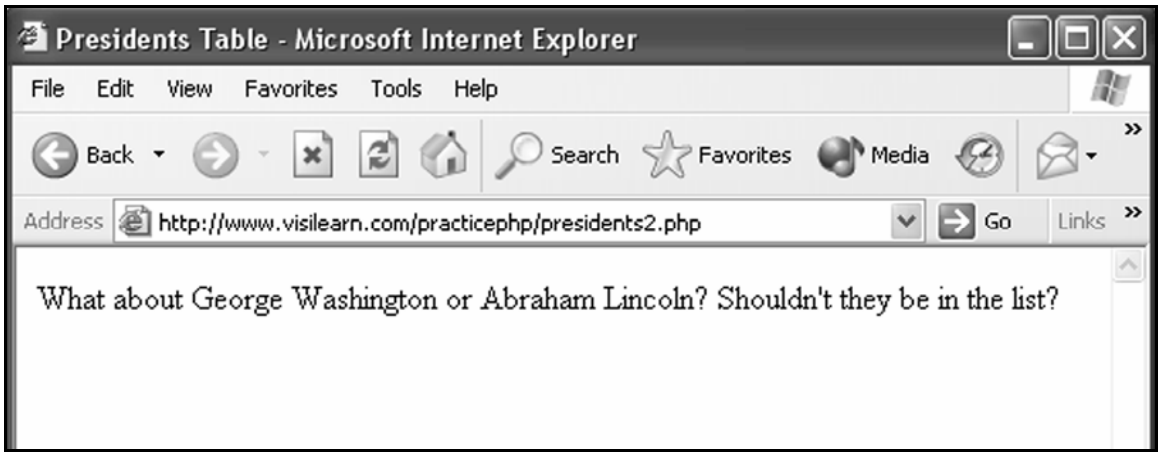
-Checks to see if the names “George Washington” or “Abraham Lincoln” were input.

-If either name was input, the script prints out all three names in an HTML table with a gray background:



-If neither name was input, it prints:

What about George Washington or Abraham Lincoln? Shouldn’t they be in the list?



7. In your browser, go to:

   www.yourwebsite.com/entry.html

8. Enter the correct username and password combination, so you're taken to **presidents.html**.

9. Test the page to make sure **presidents.php** works.

Working With Files

In this section, you'll learn how to:

- **Create a text file**
- **Display files**
- **Append to files**

Create a text file

1. Create a new Web page with this code:

```
<html>
<head>
<title>Create Text File</title>
</head>

<body>

<h2>Today's Thought</h2>

<form action="
http://www.yourwebsite.com/phpscripts/textwrite
r.php" method="post">

<input type="hidden" name="filename"
value="textthought.txt">

<textarea name="comments" rows=3 cols=50
wrap></textarea>

<p><input type="submit" value="Create Thought"
name="submit">

</form>

</body>
</html>
```

2. Save the page as **textwriter.html** in the **PHPSCRIPTS** folder, then upload it to the home directory in your Web site.

3. Create a new script with this code:

```
<?php

$myfile = fopen ($filename, 'w') or die("Can
not open file");

$outputFile = “$comments\n\n”;

fputs($myfile, $outputFile);

fclose($myfile);

print "<p>The $filename file is created with
the following thought:</p>";

print "<p>$comments</p>";

print "<p><a href=\"textwriter.html\">Enter a
new thought</a></p>\n";

print "<p><a href=\"$filename\">View the
$filename text file</a></p>\n";

?>
```

4. Save the file as **textwriter.php** in the **PHPSCRIPTS** folder.

Here's what the relevant lines in this script do:

- **`$myfile = fopen ($filename, 'w') or die("Can not open file");`**

  The form in **textwriter.html** has a hidden text field named filename. Here the script requests the value assigned to it in the form—**textthought.txt**, then assigns that value to the variable **`$myfile`**.

  The '**`w`**' means the text file **textthough.txt** (**`$myfile`**) is opened for writing.

  If the file does not exist, a new one will be created. If the file already exists, it will overwrite the existing file.

  If there is a problem with the file opening, the process will stop (or die) and display the message "Can not open file".

- **`$outputFile = "$comments\n\n";`**

  This line adds two "new line" characters (**`\n\n`**) to the end of the comments submitted through the form on **textwriter.html**.

  Any new data added to the file **textthought.txt** starts on a new paragraph. This new data is then assigned to the variable **`$outputFile`**.

- `fputs($myfile, $outputFile);`

  fputs gets the comments from the form (`$outputFile`), and puts them in the file associated with the `$myfile` variable—**textthought.txt**.

- `fclose($myfile);`

  The `fclose` command tells the Web server that you're done using the `$myfile` variable.

  Tip: *When using the* `fopen()` *command, there are six ways a file can be opened:*

  `r` *Read only.*
  `r+` *Read and write.*
  `w` *Write only.*
  `w+` *Read and over-write.*
  `a` *Append.*
  `a+` *Read and append.*

5. Upload **textwriter.php** to the **phpscripts** directory in your Web site.

6. Open **phplinks.html** and insert a new link to **textwriter.html**:

```
<p><a
href="textwriter.html">21. Create a text
file.</a></p>
```

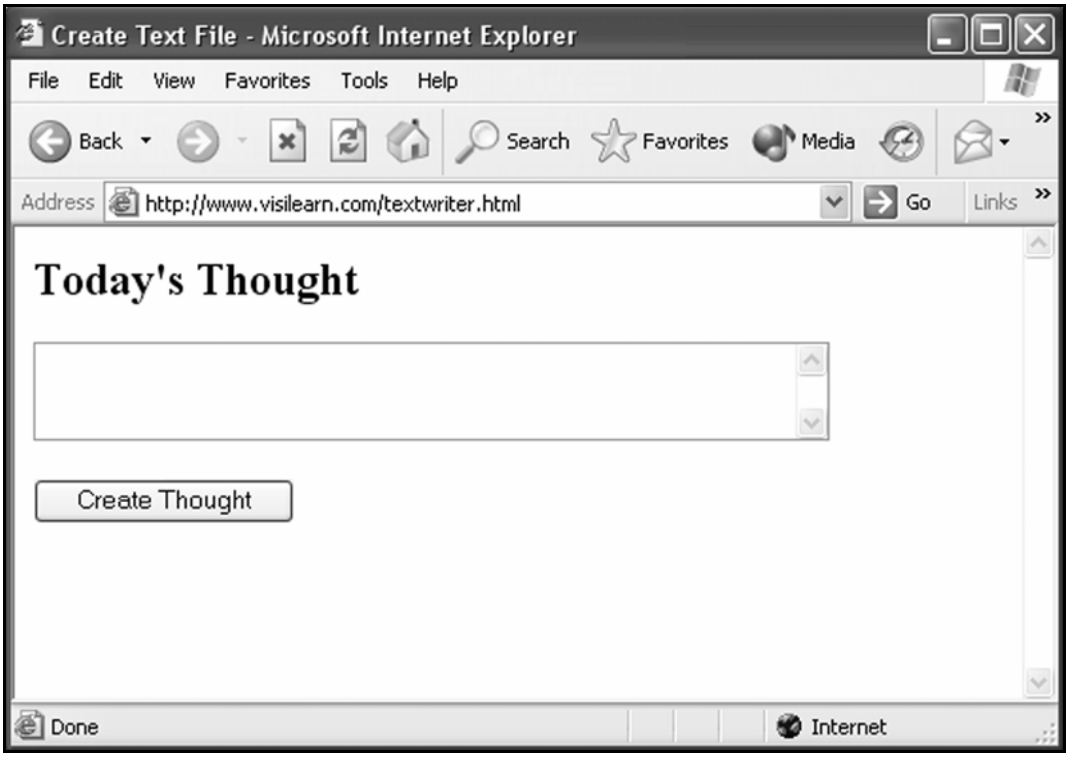
7. Save **phplinks.html**, then upload it to the home directory in your Web site.

8. In the browser, go to:

www.yourwebsite.com/phplinks.html

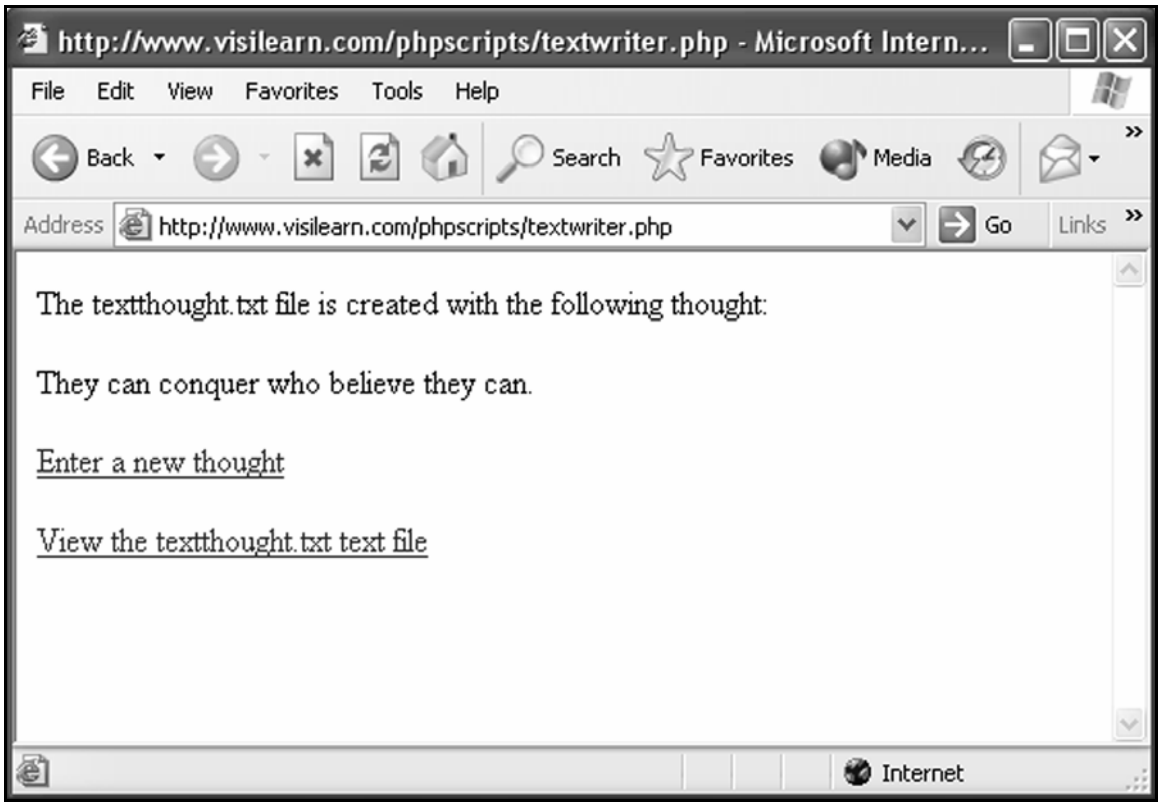
9. Click the **21. Create a text file** link.

The **textwriter.html** page will appear:



10. Type some text in the comments area, then click the Create Thought button.

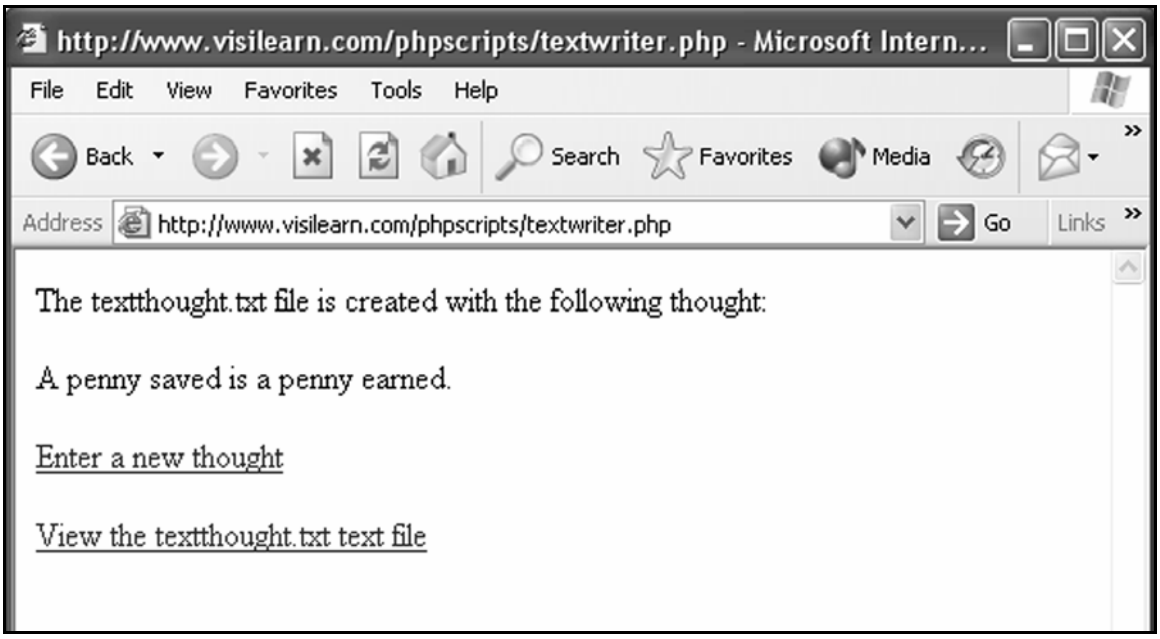
The output should look something like this:



11. Return to the **textwriter.html** page.

Enter different text in the comments area, then click the Create Thought button again.

The output should be different:



Display files

1. Create a new script with this code:

```
<?php

$myfile = fopen ('textthought.txt', 'r') or die
("Can not open file");

while (!feof($myfile)){
 $line = fgets($myfile,5000);
 print "$line <br>";
}

fclose($myfile);

?>
```

2. Save the file as **textviewer.php** in the **PHPSCRIPTS** folder, then upload it to the **phpscripts** directory in your Web site.

   Here's what the relevant lines in this script do:

   - `$myfile = fopen ('textthought.txt', 'r') or die ("Can not open file");`

     Opens the file **textthought.txt** in read-only mode, and assigns the text in it to the variable `$myfile`.

- **`while (!feof($myfile)){`**

  This **`while`** loop loops through the **`$myfile`** variable one line at a time until it gets to the end-of-file.

  `!feof` means “not the end of the file.”

  So, this line means: “Keep looping through each line in the **`$myfile`** file while it has “not” (**`!`**) reached “the end of the file” (**`feof`**).

  When the program reaches the end of the file, it will exit the loop.

- **`$line = fgets($myfile,5000);`**

  This “gets” the first line of the **`$myfile`** file and assigns it to the variable **`$line`**.

  Each time the while loop comes back to this command, it will get the next line, and then the next line, and so on.

  The **`5000`** means that each line can be up to 5000 characters long.

- **`print "$line <p>";`**

  Prints the text in each line of **textthought.txt** in a new paragraph as the script loops through them.

- `}`

  This is the end marker for the while loop.

- `fclose($myfile);`

  Once the while loop is finished, `$myfile` (the variable that stands for **textthought.txt**) is closed.

3. Open **phplinks.html** and insert a new link to **textviewer.php**:

```
<p><a
href="
http://www.yourwebsite.com/phpscripts/textviewe
r.php">22. Display a text file.</a></p>
```

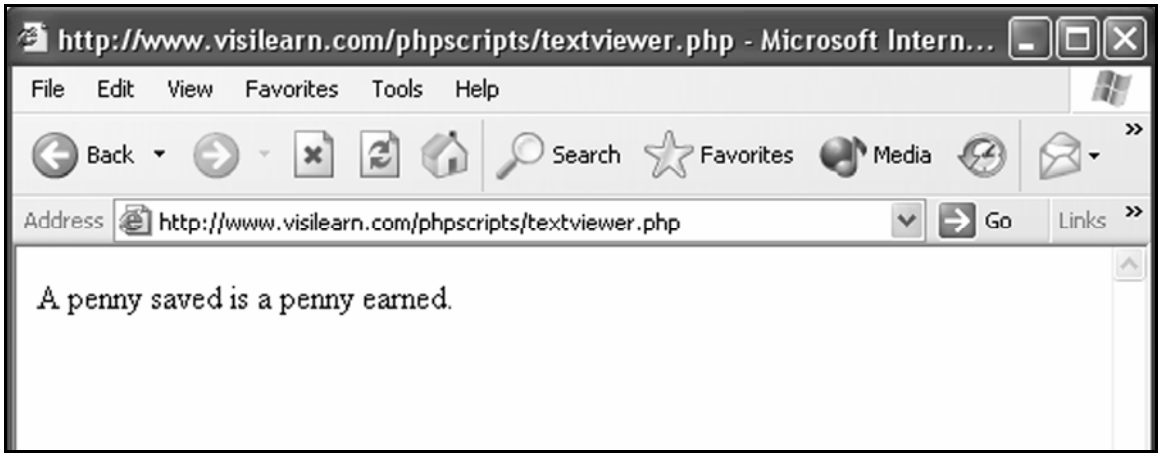
4. Save **phplinks.html**, then upload it to the home directory in your Web site.

5. In the browser, go to:

www.yourwebsite.com/phplinks.html

6. Click the **22. Display a text file** link.

The output should look like this:



Append to files

1. Create a new script with this code:

```
<html>
<head>
<title>Append to files</title>
</head>

<body>

<h2>Add to Today's Thought</h2>

<p><b>Today's Thought is ...</b></p>

<?php
$myfile = fopen ('textthought.txt', 'r') or die
("Can not open file");

while (!feof($myfile)){
 $line = fgets($myfile,4096);
 print "$line <br>";
}
fclose($myfile);
?>

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

<input type="hidden" name="filename"
value="textthought.txt">

<b>Write what you want to add here ... </b><br>

<textarea name="comments" rows=3 cols=50
wrap></textarea>

<br><input type="submit" value="Update
Thought">
```

```
</form>

</body>
</html>
```

2. Save the script as **textappenderform.php** in the **PHPSCRIPTS** folder, then upload it to the **phpscripts** directory in your Web site.

This script combines a page (**textwriter.html**) and a script (**textviewer.php**)

```
<?php
$myfile = fopen ('textthought.txt', 'r') or die
("Can not open file");

while (!feof($myfile)){
 $line = fgets($myfile,4096);
 print "$line <br>";
}
fclose($myfile);
?>
```

into one script.

It displays the text in the **texthtought.txt** file, and creates a form that allows you to add new text to the end of it.

The one change is that the form now calls a new script called textappender.php:

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

3. Create a new script with this code:

```
<?php

$myfile = fopen ($filename, 'a') or die("Can
not open file");

$outputFile = "$comments\n";

fputs($myfile,$outputFile);

fclose($myfile);

print "<p>The $filename file is updated!</p>";

print "<a href=\"$filename\">View the $filename
text file</a>\n";

?>
```

4. Save the file as **textappender.php** in the **PHPSCRIPTS** folder, then upload it to the **phpscripts** directory in your Web site.

Here's what the relevant lines in this script do:

- `$myfile = fopen ($filename, 'a') or die("Can not open file");`

  The ‘a’ parameter is used with the `fopen()` command.

  This tells the Web server to append new text to the end of the existing file, without overwriting the text already there.

5. Open **phplinks.html** and insert a new link to **textappenderform.php**:

```
<p><a
href="
http://www.yourwebsite.com/phpscripts/textappen
derform.php">23. Append new data to a text
file.</a></p>
```

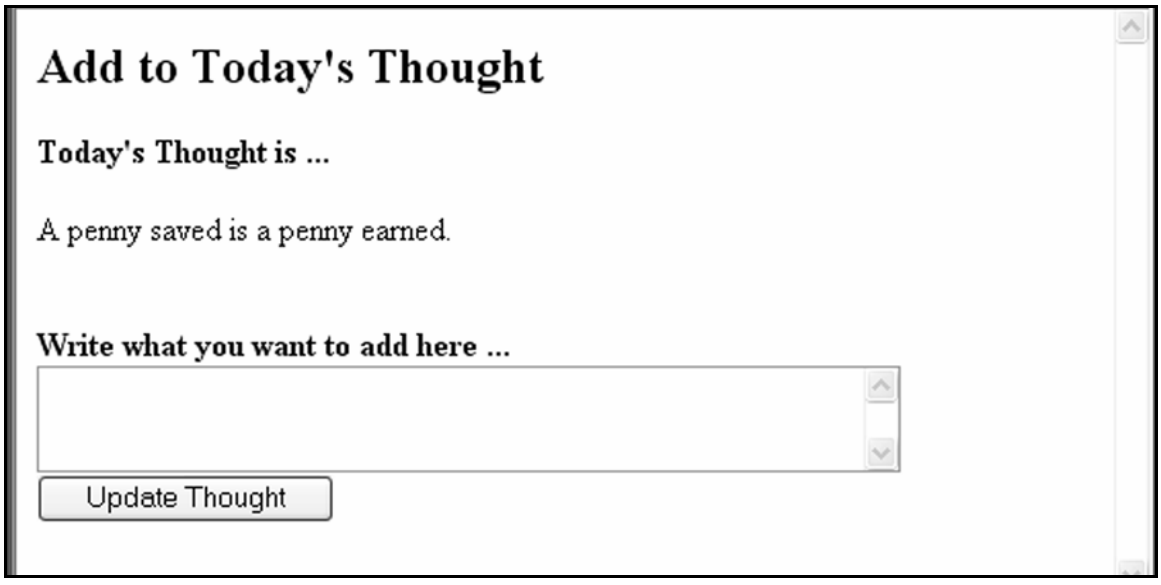
6. Save **phplinks.html**, then upload it to the home directory in your Web site.

7. In the browser, go to:

www.yourwebsite.com/phplinks.html

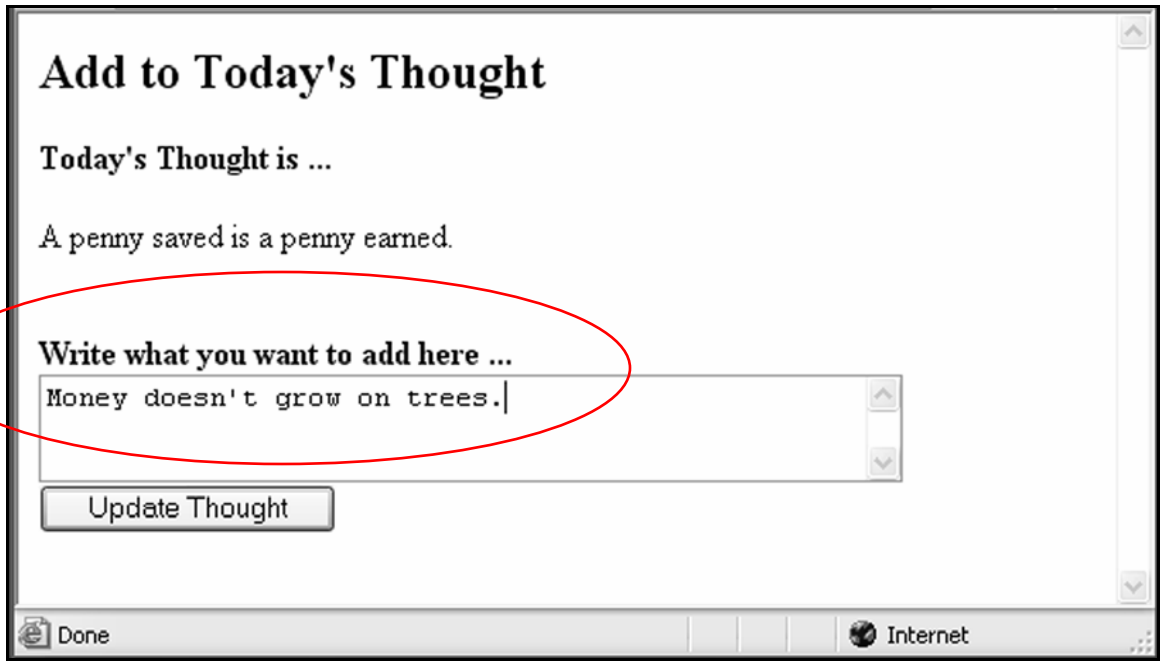
8. Click the **23. Append new data to a text file** link.

The output should look like this:



9. In its comments box, type:

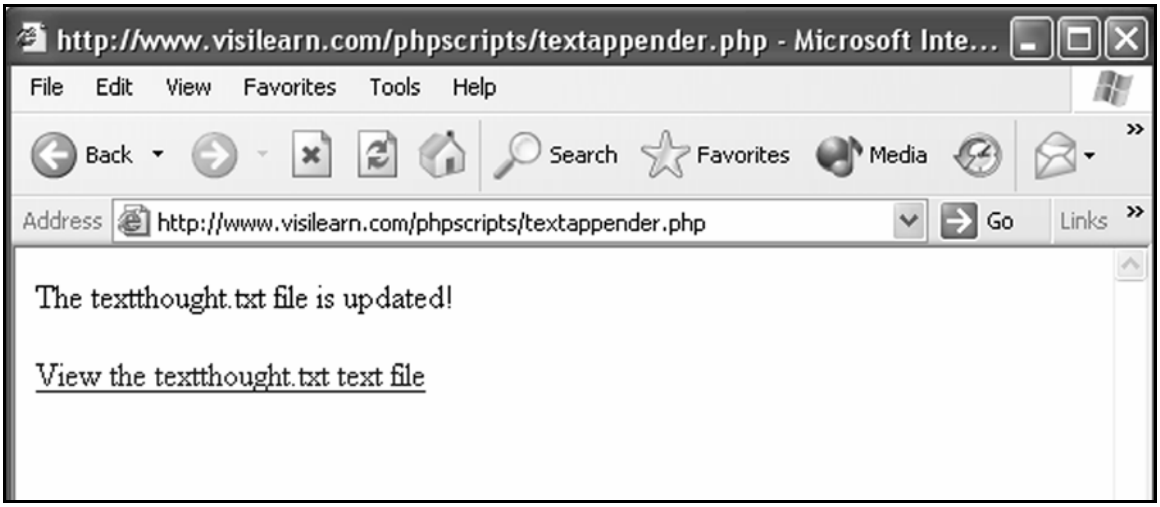
Money doesn't grow on trees.



10.

11. Click the Update Thought button.

The output should look like this:



12. In the browser, go to:

www.yourwebsite.com/phplinks.html

13. Click the **22. Display a text file** link.

The new thought should appear below the first.

Practice: Working With Files

1. Create a Web page and script, **quotes.html** and **quotes.php**, that work together to create a new text file named quotes.txt.

Make sure that **quotes.php** generates links that allow you to change and view **quotes.txt**.

2. Save **quotes.html** and **quotes.php** in the **PRACTICEPHP** folder on your computer.

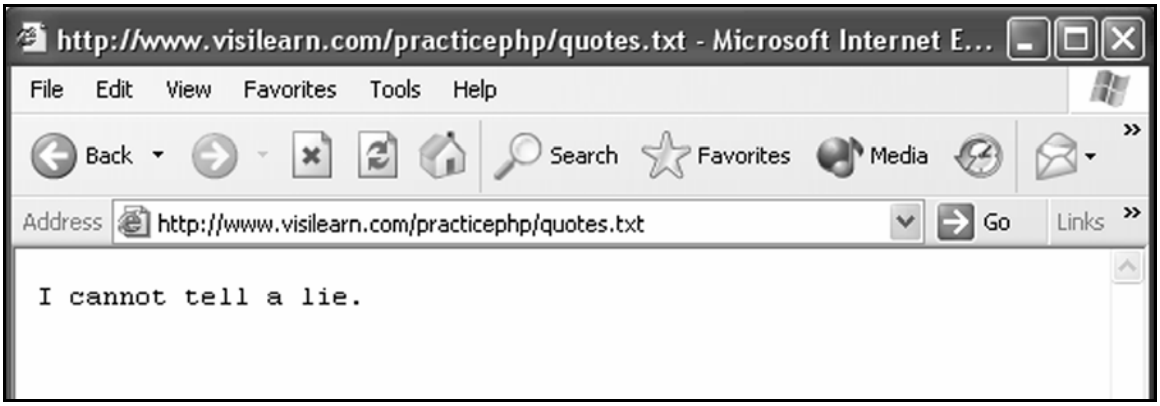
3. Upload **quotes.html** to the home directory in your Web site, then upload **quotes.php** to the **practicephp** directory in your Web site.

4. Using **quotes.html**, enter and submit this quote:

I cannot tell a lie.

5. Click the link to view quotes.txt.

It should look like this:

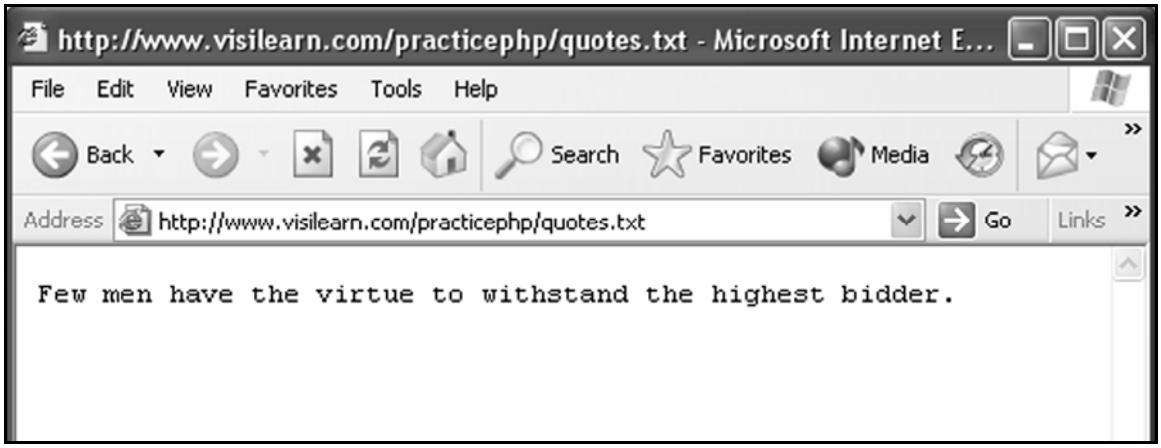


6. Enter and submit another quote:

   Few men have virtue to withstand the highest bidder.

7. Click the link to view **quotes.txt**.

   It should look like this:



8. Create a Web page and script, **append.html** and **append.php**, that work together to allow you to add more than one quote to **quotes.txt**.

9. Save **append.html** and **append.php** in the **PRACTICEPHP** folder on your computer.

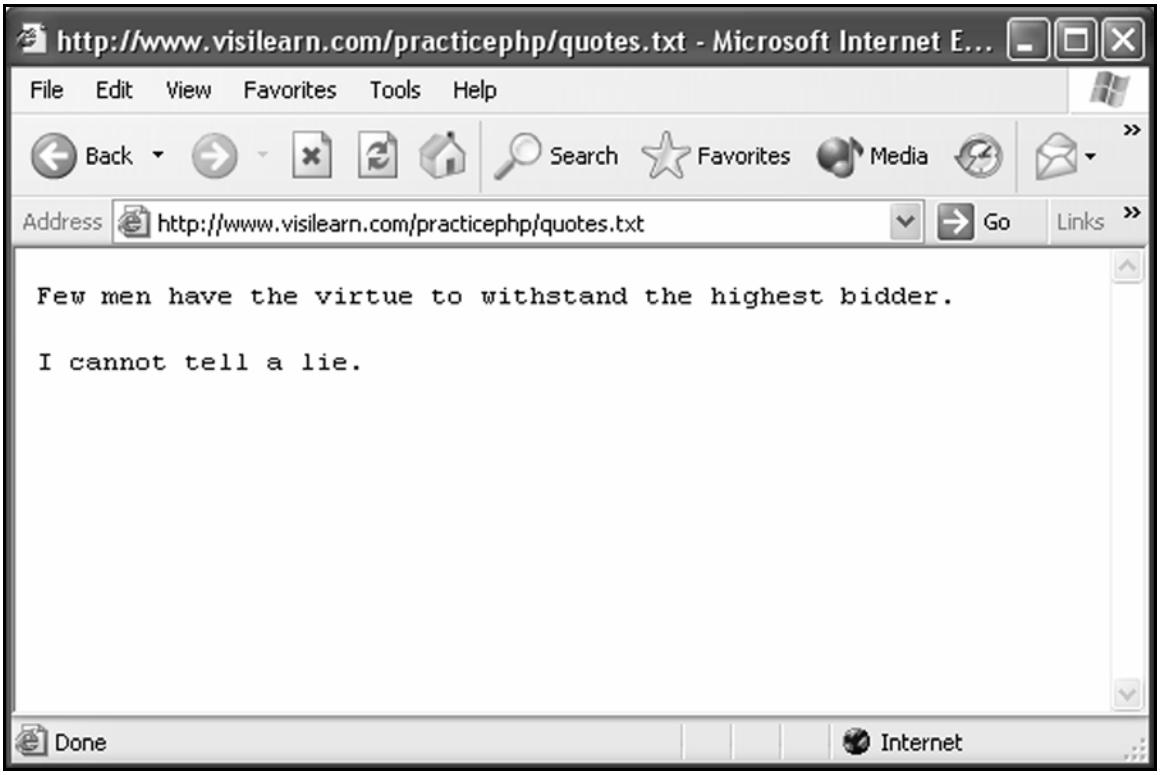
10. Upload **append.html** to the home directory in your Web site, then upload **append.php** to the **practicephp** directory in your Web site.

11. Using **append.html**, enter and submit this quote:

    I cannot tell a lie.

12. Create a script that allows you to view **quotes.txt**.

It should show both quotes:



Modifying scripts

Modifying downloaded scripts

Just about any basic script you'd need has already been written by someone else. Do a Web search on "php scripts" and you'll find many sites with good, useful scripts. Most are either free or inexpensive to use.

Now that you're familiar with PHP, you can download an existing script that generally does what you want, then modify it to meet your specific needs.

Below is an example of a script downloaded from a Web site. It takes form data from a Web page and sends it to an email address. Modify a few sections, and you can use it on your Web site.

Checks the referring form for empty form inputs. Prints an error message if any are left empty.

```
<?php

if( empty($name) || empty($email) ){
print "<html><head><title>Name and Email
Required</title></head>";
print "<body><h3>Please enter your name and
email.</h3>";
print "<h3><a href=\"mailform.html\">Click here
to
return to form.</a></h3>";
print "</body></html>";
exit;
}
```

Prevents spammers from hijacking the script, and sending messages fromthe script's URL instead of its referring form.

```
if ($_SERVER['REQUEST_METHOD'] != "POST") {
exit;
}

$headers .= "From: $name <$email>\n";
$headers .= "Reply-To: <$email>\n";
$headers .= "Content-Type: text/plain;
charset=iso-8859-1\n";
$recipient = "you@yourwebsite.com";
$subject = "Contact From Your Website";

$msg = wordwrap( $msg, 1024 );

//send the message using the
//server's mail() function

mail($recipient, $subject,
stripslashes($msg), $headers);

?>

<html>
<head>
<title>Mail Sent</title>
</head>

<body>
<h2>Thank you <? echo "$name"; ?>!</h2>
<p><strong>Your email has been sent.</strong>

</body>
</html>
```

Takes sender's name and email address from the referring form, and plugs them into the "from" and "reply to" headers of the email sent to you.

The "recipient" and "subject" of this email are specified by you.

The Web page that appears on the sender's screen after the email is sent.

Where to Get Visibooks

If you liked using this book, and would like to use more like it, visit:

www.visibooks.com

Visibooks offers more than 30 titles on subjects such as:

- **Computer Basics**
- **Microsoft Office**
- **Desktop Linux**
- **OpenOffice.org**
- **Web Site Layout**
- **Web Graphics**
- **Web Programming**

Visibooks: the simplest way to learn
and teach computer subjects.

www.visibooks.com