

WRITE T

It is often necessary to access text files stored on the Web server from within a PHP page. Writing data to text files allows you to store data, such as configuration settings and application information.

Before you can write to a file, you must establish a connection to the file by using the `fopen` function to open a file and create a file pointer. A file pointer indicates the file you want to work with. The `fopen` function takes two arguments—the name of the file to be opened and the access mode you want to use.

To open a file for writing, the `w` access mode can be specified. This access mode creates a new file if the file does not already exist. Using the `w` access mode will also overwrite any information already in an existing file.

To open a file for writing, the `w` access mode can be specified. This access mode creates a new file if the file does not already exist. Using the `w` access mode will also overwrite any information already in an existing file.

When a file is opened for writing, the file position indicator is placed at the start of a file. The file position indicator indicates where the next operation will be performed in the file.

You can use the `fputs` function to write a line of information to the file. The `fputs` function usually takes two arguments—a file pointer and the information to be written to the file. You may also specify the maximum length of the line to be written, in bytes. After the maximum length has been reached, any extra information at the end of a line will not be written to the file.

Once all the information has been written to a file, the `fclose` function should be used to close the connection to the file.

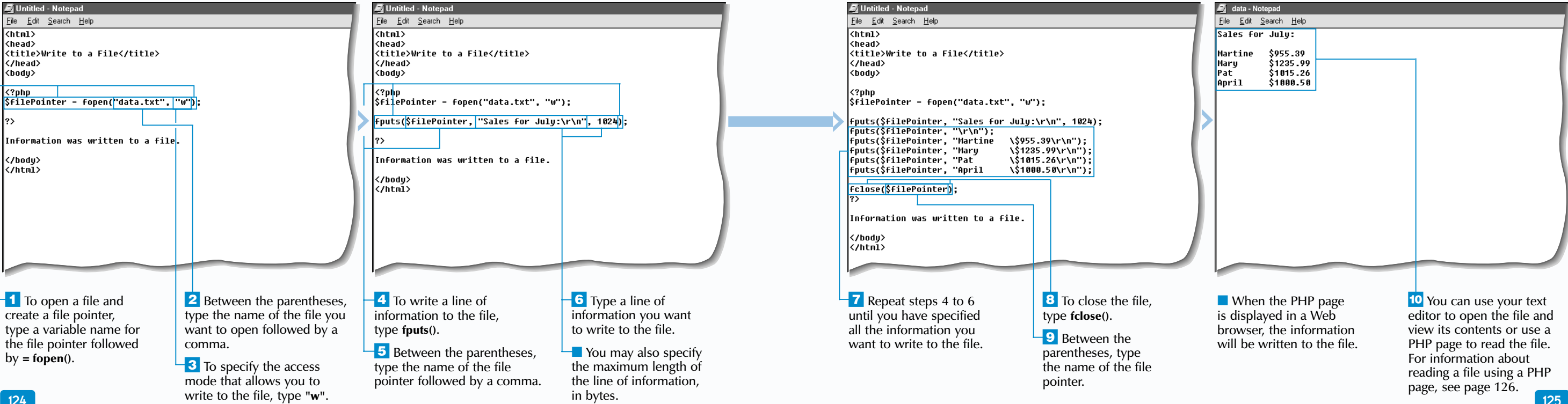
As with other operations involving accessing a file, the proper permissions that allow the file to be written to must be in place. For information about permissions, you should consult your operating system's documentation.

Extra

There are several different access modes you can use when opening a file. Each access mode places the file position indicator in a specific location.

ACCESS MODE:	DESCRIPTION:
r	Open a file for reading only. Place the file position indicator at the start of the file.
r+	Open a file for reading and writing. Place the file position indicator at the start of the file.
w	Open a file for writing only. Place the file position indicator at the start of the file. Create the file if it does not exist.
w+	Open a file for reading and writing. Place the file position indicator at the start of the file. Create the file if it does not exist.
a	Open a file for writing only. Place the file position indicator at the end of the file. Create the file if it does not exist.
a+	Open a file for reading and writing. Place the file position indicator at the end of the file. Create the file if it does not exist.

WRITE TO A FILE



READ A FILE

After creating a file pointer and opening a file for reading, you can access the contents of the file. This allows you to display the data in a Web browser or compare the data in the file to other data. When opening the file, you must specify an access mode that allows reading from the file. The `r` access mode is typically used.

When a file is initially opened for reading, the file position indicator is placed at the beginning of the file. The `fgets` function is used to read the line of text indicated by the file position indicator and must have two arguments—the file pointer and the maximum length of the line to be read, in bytes. After a line of text is read from the file, the file position indicator automatically moves to the next line of text in the file.

A `while` loop is often used to process each line in a file. With each iteration of the loop, the information retrieved from the file using the `fgets` function can be assigned to a variable and displayed to the client using the `print` function.

The `fgets` function will read a line of text up to the maximum length specified or until it reaches a newline character. A line will be truncated if it is longer than the maximum length specified. When the `fgets` function is used again, it will read the next line in the file and not the remainder of the previous line that was truncated.

The `feof` function can be used to determine if the end of a file has been reached. The `feof` function takes the file pointer as an argument and is usually used with the `Not` operator (!) as the condition tested in a `while` loop.

Once all the lines from a file have been read, the `fclose` function should be used to close the connection to the file.

Extra

Most computers that use Web server software and PHP have sophisticated username and password-based security features. When using PHP to read a file, the correct permissions that allow the PHP script to read the file must be enabled on the Web server that stores the file. Web servers are typically configured to use a special user account, such as `web` or `http`, to allow access to files. This user account must have read permissions for the files to be opened because when PHP attempts to read a file, the operating system interprets it as an attempt to read the file by the username assigned to the Web server. The special user account on the Web server must be set up before files can be accessed using PHP.

The `file` function may be used to automatically open, read and close a file. The `file` function takes the name of a file as an argument and returns the entire contents of the file as an array. Each element in the array will correspond to a line of information in the file. You do not need to create a file pointer when using the `file` function.

Example:

```
$lines = file("data.txt");
foreach ($lines as $value)
{
    print $value . "<br>";
}
```

READ A FILE

```
<html>
<head>
<title>Read a File</title>
</head>
<body>

<pre>

<?php
$filePointer = fopen("data.txt", "r");
?>

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

1 Perform steps 1 and 2 on page 124 to enter the code that opens a file and creates a file pointer.

2 To specify the access mode that allows you to read information from the file, type `"r"`.

```
<html>
<head>
<title>Read a File</title>
</head>
<body>

<pre>

<?php
$filePointer = fopen("data.txt", "r");
fgets($filePointer, 4096);
?>

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

3 To read a line of information from the file, type `fgets()`.

4 Between the parentheses, type the name of the file pointer followed by a comma.

5 Type the maximum length of the line of information to be read, in bytes.

```
<html>
<head>
<title>Read a File</title>
</head>
<body>

<pre>

<?php
$filePointer = fopen("data.txt", "r");

while (!feof($filePointer))
{
    $line = fgets($filePointer, 4096);
    print $line;
}

fclose($filePointer);

?>

</pre>
</body>
```

6 Create a `while` loop that will read and display each line of information from the file.

7 To close the file, type `fclose()`.

8 Between the parentheses, type the name of the file pointer.

Read a File - Microsoft Internet Explorer

Address http://127.0.0.1/readfile.php

Sales for July:

Martine	\$955.39
Mary	\$1235.99
Pat	\$1015.26
April	\$1000.50

9 Display the PHP page in a Web browser.

The Web browser displays the result of reading a file.

COPY OR DELETE A FILE

PHP provides functions that allow you to manage files from within a PHP page. For example, the `copy` function creates a copy of a file, while the `unlink` function deletes a file. Copying a file is useful for creating a backup copy of a file. Deleting a file lets you remove a file that is no longer needed by the PHP page.

It is good programming practice to verify whether a file exists before attempting to copy or delete the file. You can use the `file_exists` function to determine whether a file exists. The `file_exists` function returns a value of `true` if a file you specify exists and a value of `false` if the file does not exist.

The `copy` function takes two arguments—the filename or path of the file to be copied and the filename or path of the copy you want to create. When the file is copied, the `copy` function duplicates the file without removing the original file.

The `unlink` function derives its name from the command used to delete files on Unix computers. The `unlink` function takes the filename or path of the file you want to delete as its argument.

When specifying arguments for the `copy` and `unlink` functions, if the file you want to copy or delete is located in the current directory, you can specify just the name of the file. If the file is located in a different directory, you must specify the full path to the file. You may want to store the name or path of a file in a variable and then use the variable with the `copy` or `unlink` function.

When the `copy` and `unlink` functions are successful, they return an integer value other than 0. If the functions are not successful, they return a value of 0.

Extra

In order to successfully copy and delete files, you must have the appropriate file and operating system permissions on the computer where the files are stored. For example, you cannot delete a file that has read-only permissions set. For information about the permissions for an operating system, refer to the operating system's documentation.

The `copy` function is also useful for moving a file from one directory to another. To move a file, you use the `copy` function to create a copy of the file in the directory you want to move the file to. You can then use the `unlink` function to remove the original file from the directory you no longer want to store the file.

Example:

```
copy("formdata.txt", "backup/webfile/formdata.txt");
unlink("formdata.txt");
```

You can use the `filesize` function to determine the size of a file in bytes. Determining the size of a file is useful when you want to delete large files in order to free up storage space.

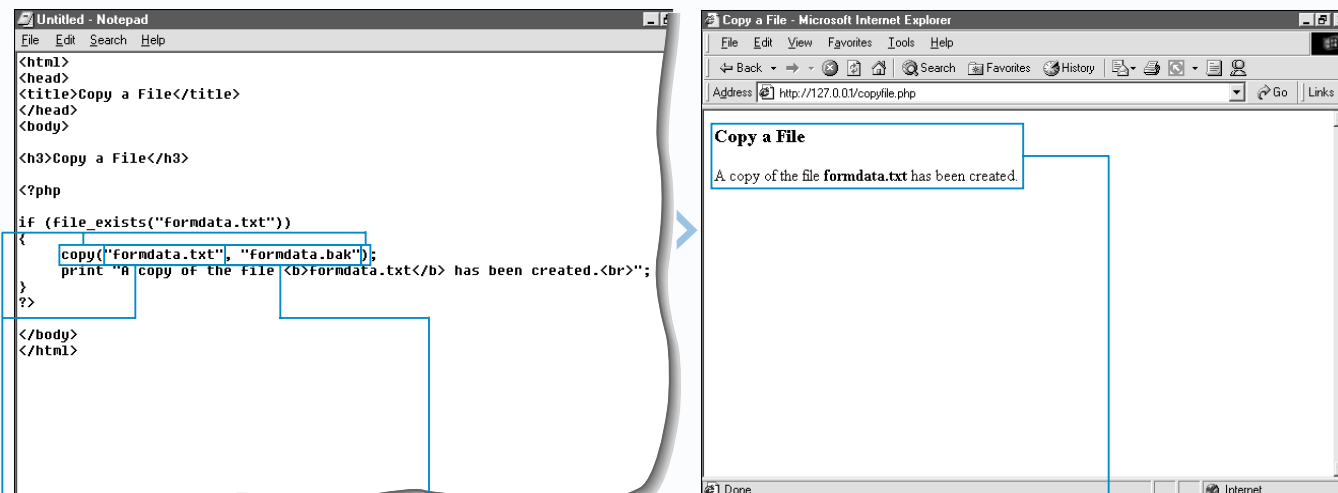
TYPE THIS:

```
print "The size of the file is: ";
print filesize("formdata.txt") . " bytes";
```

RESULT:

The size of the file is: 1867 bytes

COPY A FILE



1 To copy the file, type `copy()`.

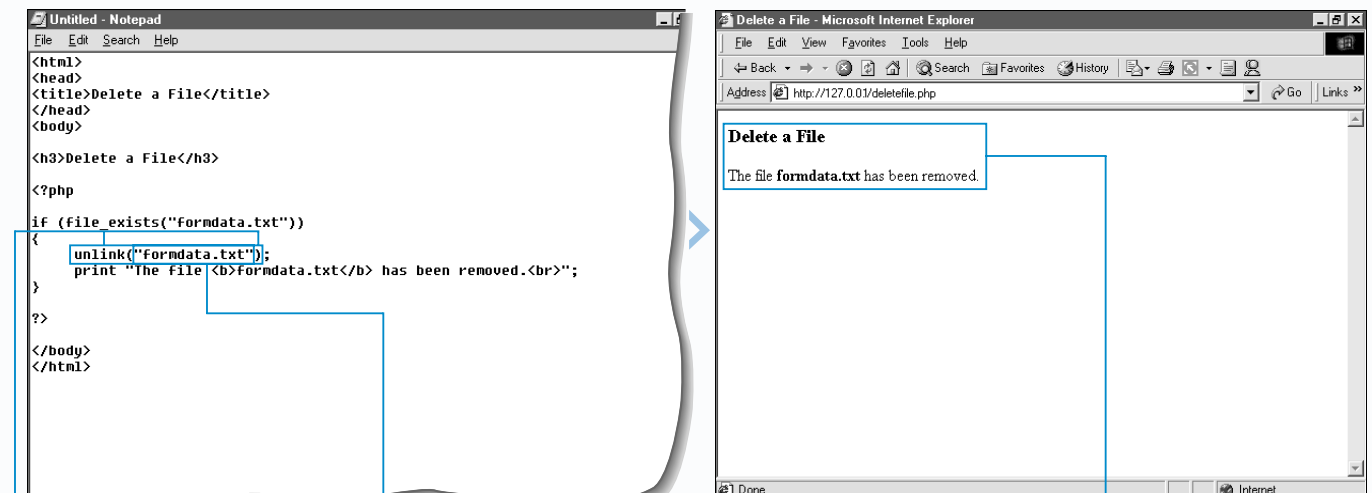
2 Between the parentheses, type the name or path of the file you want to copy enclosed in quotation marks.

3 Type a comma and then type the name or path of the copy you want to create enclosed in quotation marks.

4 Display the PHP page in a Web browser.

The file has been copied.

DELETE A FILE



1 To delete a file, type `unlink()`.

2 Between the parentheses, type the name or path of the file you want to remove enclosed in quotation marks.

3 Display the PHP page in a Web browser.

The file has been deleted.

DETERMINE THE STATUS OF A FILE

When working with files, it is often necessary to determine information about a file before performing an action. For example, you should verify that a file can be written to by a PHP page before opening the file for writing.

The `is_file` function allows you to determine if a file you specify is a file. This is useful when you want to ensure that an item is a file rather than a directory or a link to another file. The `is_file` function is commonly used to verify that an item is a file before attempting to open the file. The function returns a value of true if the item you specify is a file.

Before opening a file you want to read, it is common practice to use the `is_readable` function to verify that the file can be read from a PHP page. If a file exists and is readable, the `is_readable` function returns a value of true.

To determine whether a file can be written to by a PHP page, you use the `is_writable` function. If the file exists and can be written to, the function returns a value of true. The `is_writable` function is commonly used to check the status of files that can be written to by multiple users, since such files are frequently temporarily unavailable for writing while in use by another user.

The `is_file`, `is_readable` and `is_writable` functions all take a single argument that indicates the file you want to check. The argument can be the file name or path of the file, such as `c:\webfiles\form.dat`, or a variable that stores the filename or path, such as `$filename`. If the file you want to check is in the current directory, you can specify just the name of the file. If the file is located in a different directory, you must specify the full path of the file.

Extra

Determining the status of a file can improve the efficiency of a PHP script. For example, before performing a complex process, such as retrieving information from a database and appending the information to a file, a check can be made to ensure the data can be written to the file.

The `is_file`, `is_readable` and `is_writable` functions may be affected by the operating system you are running and the file system used by the computer that stores the files. While the `is_file`, `is_readable` and `is_writable` functions should perform as expected on Unix systems, there may be incompatibilities that interfere with the operation of these functions on other systems.

You can use the `is_dir` function to determine whether an item is a directory. The `is_dir` function returns a value of true if the directory you specify exists and is a directory.

TYPE THIS:

```
if (is_dir("c:\data\webinfo"))
{
    print "c:\data\webinfo is a directory.";
}
else
{
    print "This is not a directory.";
}
```

RESULT:

c:\data\webinfo is a directory.

DETERMINE THE STATUS OF A FILE

```
<html>
<head>
<title>Determine File Status</title>
</head>
<body>

<h3>Determine File Status</h3>

<?php
$fileName = "test.php";

if (is_file($fileName))
{
    print "$fileName is a file.<br>";
}
else
{
    print "$fileName is not a file.<br>";
}

?>

</body>
</html>
```

1 To store the path of a file you want to check in a variable, type the code that assigns the path to the variable.

2 To determine whether the file is a file, type `is_file()`.

3 Between the parentheses, type the name of the variable that stores the path of the file.

4 Type the code that uses the result of determining whether the file is a file.

```
<h3>Determine File Status</h3>

<?php
$fileName = "test.php";

if (is_file($fileName))
{
    print "$fileName is a file.<br>";
}
else
{
    print "$fileName is not a file.<br>";
}

if (is_readable($fileName))
{
    print "$fileName is readable.<br>";
}
else
{
    print "$fileName is not readable.<br>";
}

?>
```

5 To determine whether the file can be read from the PHP page, type `is_readable()`.

6 Between the parentheses, type the name of the variable that stores the path of the file.

7 Type the code that uses the result of determining whether the file is readable.

```
<h3>Determine File Status</h3>

<?php
$fileName = "test.php";

if (is_file($fileName))
{
    print "$fileName is a file.<br>";
}
else
{
    print "$fileName is not a file.<br>";
}

if (is_readable($fileName))
{
    print "$fileName is readable.<br>";
}
else
{
    print "$fileName is not readable.<br>";
}

if (is_writable($fileName))
{
    print "$fileName can be written to.<br>";
}
else
{
    print "$fileName cannot be written to.<br>";
}

?>
```

8 To determine whether you can write to the file, type `is_writable()`.

9 Between the parentheses, type the name of the variable that stores the path of the file.

10 Type the code that uses the result of determining whether the file is writable.

```
Determine File Status - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Home Search Favorites History
Address http://127.0.0.1/determinestatus.php Go Links
Done Internet

Determine File Status
test.php is a file.
test.php is readable.
test.php can be written to.
```

11 Display the PHP page in a Web browser.

The Web browser displays the results of determining the status of a file.

SET FILE GROUP AND PERMISSIONS ON UNIX

You can use a PHP page to change the group a file belongs to and set the permissions for the file. Changing the group and permissions for a file allows you to restrict access to the file.

The `system` function allows you to execute UNIX commands from within a PHP page. To view the group and permissions for each file in the current directory, you can use the `ls -l` command.

The `chgrp` function is used to assign a file to a new group. To use the `chgrp` function, you specify the location and name of the file you want to assign to a new group followed by the name of the group.

Unix systems allow you to assign any combination of read (r), write (w), and execute (x) permissions to a file's owner, the group the file belongs to and all other users. The permissions for a file are typically represented by a

dash followed by a series of nine characters. The first three characters represent the permissions for the file's owner. The next three characters represent the permissions for the owner's group and the last three characters represent the permissions for all other users.

To change the permissions for a file, you use the `chmod` function. The `chmod` function takes two arguments—the location and name of the file and the octal value representing the file permissions you want to use. The octal value consists of three digits, which represent the permissions for the owner, followed by the permissions for the owner's group and the permissions for other users. For example, the `-rwxr-xr-x` permissions will have an octal value of 755. You should precede the octal value with a zero (0).

The `chgrp` and `chmod` functions return a value of true when the functions are executed successfully.

Extra

Only the owner of a file or the system administrator can change the group or permissions for a file. When using the `chgrp` function, the owner of a file can assign the file only to a group to which the owner belongs. The system administrator can assign a file to any group.

The `chown` function can be used to change the owner of a file. When using the `chown` function, you must specify the location and name of the file and the user name of the new owner. Only the system administrator can use the `chown` function.

Example:

```
chown("/var/www/html/file1", "maryc")
```

The `fileowner` function allows you to determine the owner of a file. You can use the `filegroup` function to determine the group a file belongs to, which is usually the group to which the owner belongs. These functions retrieve an ID number that represents the user name of the owner or the name of the group. You can use the password file or group file available in Unix to match an ID number with the corresponding user name or group name.

TYPE THIS:

```
print ("The UID is: " . fileowner("/var/www/html/file1") . "br");
print ("The GID is: " . filegroup("/var/www/html/file1"));
```

RESULT:

The UID is: 48
The GID is: 501

SET FILE GROUP AND PERMISSIONS ON UNIX

```

<html>
<head>
<title>Set File Permissions</title>
</head>
<body>

<?php
print "<b>Old file permissions:</b><br>";
print "<pre>";
system("ls -l");
print "</pre>";
print "<b>New file permissions:</b><br>";
print "<pre>";
print "</pre>";
?>

```

1 To execute a Unix command from the PHP page, type `system()`.

2 To display a list of files in the current directory and the permissions for each file, type `"ls -l"` between the parentheses.

```

<html>
<head>
<title>Set File Permissions</title>
</head>
<body>

<?php
print "<b>Old file permissions:</b><br>";
print "<pre>";
system("ls -l");
print "</pre>";
chgrp("/var/www/html/file1", "apache");
print "<b>New file permissions:</b><br>";
print "<pre>";
print "</pre>";

```

CHANGE FILE GROUP

3 To assign a file to a new group, type `chgrp()`.

4 Between the parentheses, type the location and name of the file, enclosed in quotation marks.

5 Type a comma, followed by the name of the group to which you want to assign the file, enclosed in quotation marks.

```

</head>
<body>

<?php
print "<b>Old file permissions:</b><br>";
print "<pre>";
system("ls -l");
print "</pre>";
chgrp("/var/www/html/file1", "apache");
chmod("/var/www/html/file1", 0755);
print "<b>New file permissions:</b><br>";
print "<pre>";
system("ls -l");
print "</pre>";

```

SET FILE PERMISSIONS

6 To change the permissions for a file, type `chmod()`.

7 Between the parentheses, type the location and name of the file, enclosed in quotation marks.

8 Type a comma, followed by the octal value that represents the permissions you want to assign to the file. The octal value should be preceded by 0.

9 To display an updated list of the files in the current directory, repeat steps 1 and 2.

```

Set File Permissions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Search Favorites History
Address http://192.168.0.3/test.php Go Links

Old file permissions:
total 16
-rwxrwxrwx 1 apache sales 30 Apr 26 15:30 file1
-rw-r--r-- 1 apache apache 381 Apr 27 11:55 test.php
-rwxr-xr-x 1 root root 161 Apr 27 11:24 test2.php
-rw-r--r-- 1 root root 37 Apr 27 09:56 text.php

New file permissions:
total 16
-rwxr-xr-x 1 apache apache 30 Apr 26 15:30 file1
-rw-r--r-- 1 apache apache 381 Apr 27 11:55 test.php
-rwxr-xr-x 1 root root 161 Apr 27 11:24 test2.php
-rw-r--r-- 1 root root 37 Apr 27 09:56 text.php

```

10 Display the PHP page in a Web browser.

The Web browser displays the results of changing the group and permissions for a file.

CREATE AND DELETE DIRECTORIES

PHP provides functions that allow you to manage directories from within a PHP page. The `mkdir` function creates a new directory, while the `rmdir` function removes an existing directory. You may want to create a directory to store temporary files and then remove the directory when the files are no longer needed by the PHP page.

When working with directories in the current directory, you specify the name of the directory you want to create or remove. If you want to work with directories in a different directory, you must specify the full path of the directory you want to create or remove. You may want to store the name or path of a directory in a variable and then use the variable with the `mkdir` or `rmdir` function.

When creating a new directory on a computer running the UNIX operating system, you must also specify permissions for the directory. Permissions control access to the directory and determine the operations, such as reading and writing, that can be performed.

A directory you want to remove must not contain any files or subdirectories. You must delete any existing files and subdirectories from a directory before you can delete the directory. For information about deleting files, see page 128.

When the `mkdir` and `rmdir` functions successfully create or remove a directory, they return an integer value other than 0. If the functions are not successful, they return a value of 0.

It is good programming practice to verify whether a directory exists before attempting to create or remove the directory. You can use the `file_exists` function to determine whether a file exists.

Extra

PHP allows you to specify permissions for a directory in octal notation. When creating a directory on a computer running the UNIX operating system, you can convert UNIX permissions to octal notation using the following chart. The UNIX permissions include read (r), write (w) and execute (x). The octal value will consist of three digits, representing the permissions for the owner, followed by the permissions for the owner's group and the permissions for other users. For example, a directory with permissions `-rwxrw-r--` will have the octal value 764. When entering the permissions in a PHP script, precede the octal value with a zero (0).

UNIX PERMISSIONS:	OCTAL VALUE:
---	0
--x	1
-w-	2
-wx	3
r--	4
r-x	5
rw-	6
rwx	7

In order to successfully create and remove directories, you must have the appropriate file and operating system permissions on the computer where the directories are to be created or removed. For information about the permissions for an operating system, refer to the operating system's documentation.

The `chdir` function allows you to change which directory you are working in. This is useful when you want to work with files stored in another directory. To change the current directory, specify the name of the directory you want to work in, enclosed in quotation marks.

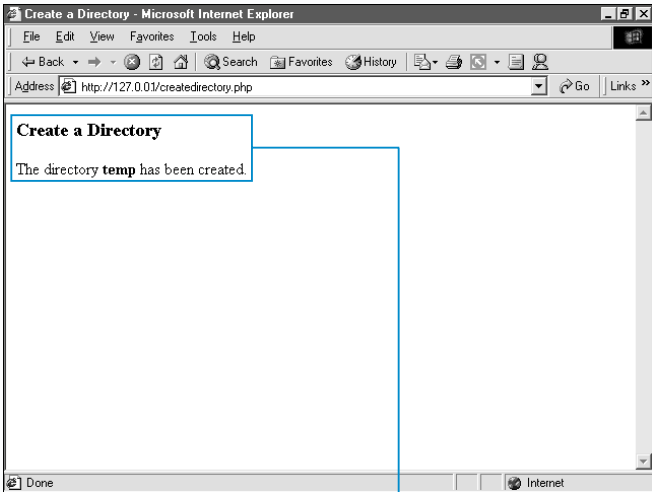
Example:
`chdir("temp");`

CREATE A DIRECTORY

```
<?php
$dirName = "temp";
if (!file_exists($dirName))
{
    mkdir($dirName, 0700);
    print "The directory <b>$dirName</b> has been created.<br>";
}
?>
```

- 1 To store the name or path of the directory you want to create in a variable, type the code that assigns the information to the variable.
- 2 To create a directory, type `mkdir()`.

- 3 Between the parentheses, type the name of the variable you created in step 1.
- 4 If necessary, type a comma and then type the octal value for the permissions you want the directory to use.

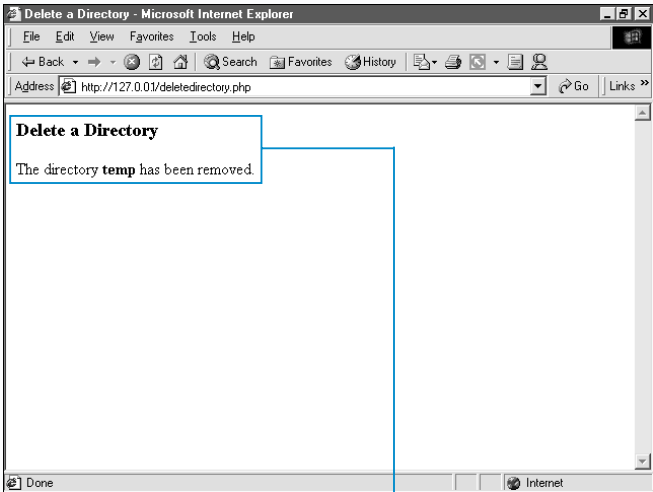


- 5 Display the PHP page in a Web browser.
- The directory has been created.

DELETE A DIRECTORY

```
<?php
$dirName = "temp";
if (file_exists($dirName))
{
    rmdir($dirName);
    print "The directory <b>$dirName</b> has been removed.<br>";
}
?>
```

- 1 To store the name or path of the directory you want to remove in a variable, type the code that assigns the information to the variable.
- 2 To remove a directory, type `rmdir()`.
- 3 Between the parentheses, type the name of the variable you created in step 1.



- 4 Display the PHP page in a Web browser.
- The directory has been removed.