

第6章 用PHP操作目录和文件

- 目录和文件是程序开发中的一个重要概念。文件通常是指，存储在计算机硬件介质（如硬盘）上的数据的集合。操作系统以文件为单元，对数据进行管理。这意味着，如果要找到计算机系统中的某些数据，必须按文件名找到指定的文件，然后才能从文件中获取有关数据。如果需要将一些数据A 舢郊扑慷 低车拇嫫 13.槽手校 脖瓿脾冉 (10) 募 拍芟蚱湫慈胍 嫫 5.氛 荨
- 以前各章所用到的输出，都是以浏览器为对象的，即向浏览器端输出数据。对用 PHP 开发的Web程序来说，输入一般都是从HTML表单提交过来的数据。本章讲述如何从文件获取数据，以及如何将数据存入文件，即程序数据的输入和输出都是通过文件来完成。



6.1 目录操作

- 目录是计算机文件系统的重要组成部分，可以将其看成是一种特殊的文件。本节介绍一些和目录操作有关的PHP内置函数。



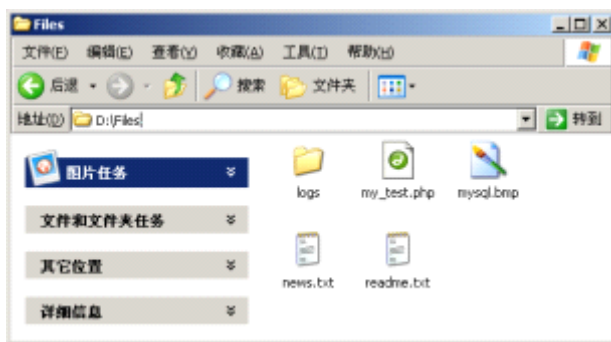
6.1.1 打开和关闭目录

- 要完成对目录的操作，首先要获取操作的目录句柄，即打开一个目录。所谓句柄，读者可以将其理解成目录的别名，或者理解成句柄代表的就是目录本身。PHP提供函数`opendir`完成打开目录的操作，该函数的语法如下所示。
- `$dh = opendir(string $path);`
- 参数`$path`是要打开的目录。如果处理成功，函数返回目录的句柄，否则函数返回`FALSE`。如果`$path`不是一个合法的目录或因为权限、文件系统等错误而无法打开目录，该函数也会返回`FALSE`。
- 对一个目录的操作结束之后，需要关闭已打开的目录，即释放操作目录时所占用的资源。PHP提供函数`closedir`完成关闭目录的操作，该函数的语法如下所示。
- `void closedir($dh);`
- 该函数关闭由函数`opendir`打开的目录句柄`$dh`，它没有返回值。这两个函数的应用演示将在下小节举例。

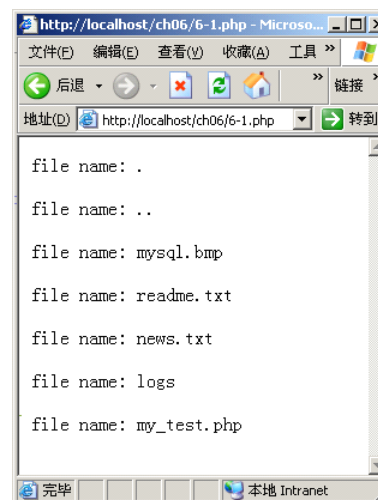


6.1.2 获取目录中下一个文件的文件名

- 使用函数readdir可以获取目录中下一个文件的文件名。该函数的语法如下所示。
- `string readdir($dh);`
- 参数\$dh是由函数opendir打开的目录句柄，该函数返回这个句柄所指目录中的文件名，如果失败，返回FALSE。



目录结构示意图



使用 readdir 获取目录中的文件名



6.1.3 列出某一目录下的所有文件

- 除了使用函数readdir可以获取目录下的文件名之外，函数 scandir也可以列出指定目录中的文件和目录，该函数是 PHP5新增的函数。它的语法如下所示。
- array scandir(string \$directory);
- 参数\$directory是指定的目录，函数返回值是一个含有文件名的数组，如果失败，函数返回FALSE。如果参数\$directory不是一个目录，函数也会返回FALSE。该函数还有一个常用的可选参数作为第2个参数，用来为文件名做排序。



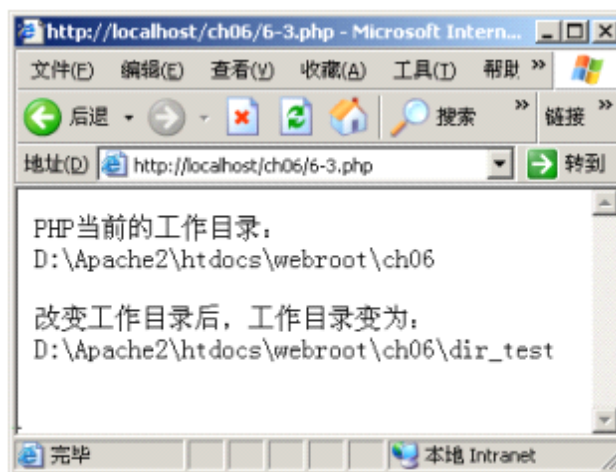
```
Array
(
    [0] => .
    [1] => ..
    [2] => news.txt
    [3] => other
    [4] => read.txt
)
Array
(
    [0] => read.txt
    [1] => other
    [2] => news.txt
    [3] => ..
    [4] => .
)
```

使用 scandir 列出目录下所有文件



6.1.4 获取和改变PHP的当前工作目录

- PHP提供以下两个函数，分别完成获取和更改 PHP当前工作目录的功能。
- `getcwd`: 该函数返回当前工作目录，如果失败返回 `FALSE`。该函数不需要传入参数。
- `chdir`: 改变PHPd的当前工作目录，该函数的参数是，要改变的工作目录。处理成功该函数返回 `TRUE`，否则返回`FALSE`。



获取和改变 PHP 当前的工作目录



6.2 文件操作

- 文件操作是程序开发中最基本的内容之一。实际应用中，经常需要从文件取数据，或者向文件写入数据，如分析日志数据和记录日志等。因此，作为编程的基本功之一，读者需要努力掌握和文件处理的有关内容，并在实际应用中加强实践。本节将为读者介绍一些基本的文件操作函数。



6.2.1 打开和关闭文件

- 在程序中，处理文件的一般流程如图 6.5所示。
- 要通过PHP读出文件内容，或者向文件写入数据，首先要做的是打开文件。在PHP中使用函数fopen打开一个文件，该函数的语法如下所示。
- `$fp = fopen(string $filename, string $mode);`
- 函数fopen可以打开本地或远程文件，参数 \$filename 是指要打开的文件的文件名，返回值是文件处理句柄 \$fp，程序通过\$fp来操作文件。如果打开失败，函数返回 FALSE。该函数的第二个参数 \$mode，它是字符串类型，指打开文件的模式。打开文件时，之所以要指定模式，是为了告诉操作系统，如何处理即将打开的文件。例如，如果需要向某文件写入一些数据，可以向该函数的第 2 个参数传参数 'w'，此时，这个参数表示的含义是，以写入方式打开文件。



6.2.2 读出文件的内容

- 文件打开后，就可以从文件中读出内容，以供程序使用。本节通过一具体实例，来讲述 PHP中和读文件有关的函数。
- （1）建立一个文本文件，供程序打开和读其内容之用。该文本内容如下所示。
- **<h3>最新动态</h3>**
- 40名企业家自费植树
- 中国战机南疆出击
- 姚明31分火箭胜太阳
- （2）将该文件以名称 info.dat 保存在服务器文档根目录的上级目录data下。下面编写读取这个文件内容的 PHP 程序，



6.2.3 向文件写入内容

- 除了从文件读出内容，有时还需要向文件写入数据。这小节就介绍如何在PHP程序中，向文件写入数据。可以通过以下3个函数完成对文件的写入操作：函数 `fwrite()`；函数 `fputs()`；函数 `file_put_contents()`。
- (1) 函数 `fwrite()` 的语法如下所示。
- `int fwrite($fp, string $content [, int $length]);`
- (2) 函数 `fputs()` 是函数 `fwrite()` 的别名，其实它就是函数 `fwrite()`，用法和 `fwrite()` 一样，这里不再赘述。
- (3) 函数 `file_put_contents()` 也可以完成向文件写入数据，该函数的语法如下所示。
- `int file_put_contents(string $filename, string $data);`



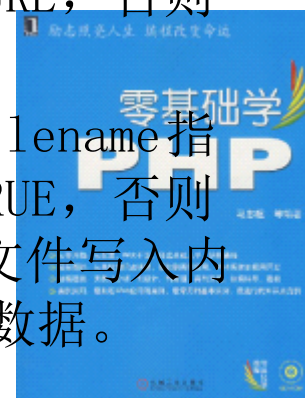
6.2.4 获取文件的相关信息

- 文件的相关信息主要涉及到，文件的所有者、文件的大小和文件的类型等。PHP可以通过以下函数获取文件的一些主要信息。
- 函数fileowner()，取得文件的所有者，返回文件所有者的用户ID，如果失败，返回FALSE。
- 函数filesize()，取得文件的大小，返回值是文件大小的字节数，如果失败，返回FALSE。
- 函数filetype()，返回文件类型，可能的值有file（文件）、dir（目录）、link（符号链接）、block（块）、unknown（未知）等。如果失败，返回FALSE。



6.2.5 一些判断文件性质的函数

- PHP中判断文件性质的函数主要有以下几个。
- 函数`is_dir($filename)`，判断由参数`$filename`指定的文件是否是目录，如果文件名存在并且是一个目录，该函数返回TRUE，否则返回FALSE。
- 函数`is_file($filename)`，判断由参数`$filename`指定的文件是否是普通文件。如果文件名存在并且是一个普通文件，该函数返回TRUE，否则返回FALSE。
- 函数 `is_readable($filename)`，判断由参数`$filename`指定的文件是否可读。文件可读时，该函数返回 TURE，否则返回FALSE。
- 函数`is_writable($filename)`，判断由参数`$filename`指定的文件是否可写。文件可写时，该函数返回 TRUE，否则返回FALSE。函数`is_writable()`经常在向一个文件写入内容之前调用，如果该文件可写，才向文件里写入数据。



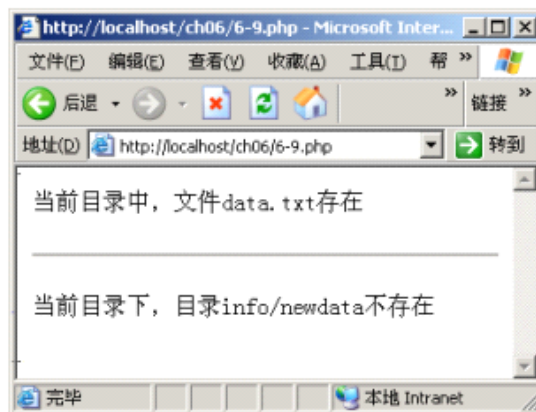
6.2.6 获取路径中的文件名和目录名

- (1) 调用函数**basename()**可以取得一个路径中的文件名部分，该函数的语法如下所示。
- `string basename(string $path);`
- 参数**\$path**是完整的路径名，该函数返回指定路径**\$path**中的文件名部分。在Windows系统中，可以使用/或\作为目录分隔符，在UNIX系统中，使用/作为目录分隔符。
- (2) 调用函数**dirname()**可以取得一个路径中的目录名部分，该函数的语法如下所示。
- `string dirname(string $path);`
- 参数**\$path**是完整的路径名，该函数返回指定路径**\$path**中的目录名。



6.2.7 判断文件或目录是否存在

- 调用函数file_exists可以检查指定的文件或目录是否存在，该函数的语法如下所示。
- `bool file_exists(string $filename);`
- 该函数判断有参数\$filename指定的文件或目录是否存在，如果存在，函数返回TRUE，否则函数返回FALSE。



使用 PHP 函数判断文件和目录是否存在



6.2.8 用PHP建立目录和删除目录

- PHP提供了在文件系统中建立目录和删除目录的函数，在PHP程序中可以很方便地使用它们。这些函数如下所示。
- 函数`mkdir($pathname, $mode)`，用来创建一个目录。参数`$pathname`是指要创建的目录的路径或名称，参数`$mode`是指所建目录的访问权限，它通常是指UNIX系统下的访问权限。如果目录创建成功，函数返回TRUE，否则返回FALSE。
- 函数`rmdir($pathname)`，用来删除一个目录。参数`$pathname`是指要删除的目录，但该目录必须是空目录，并且在UNIX下，该目录要有相应的删除权限。如果目录删除成功，函数返回TRUE，否则返回FALSE。



6.2.9 用PHP拷贝、删除和移动文件

- PHP中拷贝、删除和移动文件，也可以通过其提供的预定义函数完成。下面列举了拷贝、删除和移动文件的函数。
- 函数copy(\$source, \$desc)，用来把文件从参数 \$source 指定的位置，拷贝到由参数 \$desc 指定的位置。如果拷贝成功，该函数返回 TRUE，否则该函数返回 FALSE。
- 函数unlink(\$file)，用来删除由参数 \$file 指定的文件。如果删除成功，该函数返回 TRUE，否则返回 FALSE。
- 函数rename(\$oldname, \$newname)，用来把由参数 \$oldname 指定的文件重命名为由参数 \$newname 指定的名字。这两个参数都可以是一个带文件名的完整路径，从而完成文件从一个目录到另一个目录的移动。



6.2.10 用PHP处理文件的锁定

- 如果一个文件被某个用户的程序打开写入一些内容，同时，另外一个用户也打开这个文件，向其中写入数据，那么，文件的最终内容会是怎样的？哪个用户的程序写入的数据，将被文件系统保存？还是两个写入的数据都会被保存？或者都没有被保存？在实际应用中，经常会碰到两个或多个没C绦颖 倍列匆桓鑫募壤嗨频奈侍狻
- 为了解决这样的问题，保证文件中数据的完整有效，在操作文件时，需要对文件进行加锁操作。PHP通过函数flock完成对文件的锁定操作。该函数的语法如下所示。
- `bool flock($fp, int $operation [,int &wouldblock]);`

函数flock() 锁定种类参数\$operation的取值

锁定种类参数 \$operation	含义
LOCK_SH	读锁定，也叫共享锁定，针对读取的程序。文件可以被其它读取者共享。
LOCK_EX	写锁定，也叫独享锁定，针对写入的程序。文件不能被共享。
LOCK_UN	释放锁定，无论共享与独占。
LOCK_NB	防止锁定时阻塞。



6.2.11 用PHP更改文件的属性

- 更改文件的属性，主要涉及到更改文件的模式、所有者和更改文件的组。通常，在UNIX系统下讨论这些问题才更有意义。下面简单介绍一下相关函数。
- 函数chgrp(\$filename, \$group)，用来将文件\$filename所属的组改为\$group（可以使用组名或组ID）。如果更改成功，函数返回TRUE，否则返回FALSE。
- 函数chmod(\$filename, \$mode)，将文件\$filename的模式改为\$mode。这里的模式是指可读、可写和可执行。例如，在UNIX系统中，如果文件的模式为0777，表示所有用户对该文件可读、可写并且可执行。下面是该函数的一个示例代码。这行代码将文件/home/log/del.sh改为所有人可读、可写、可执行。

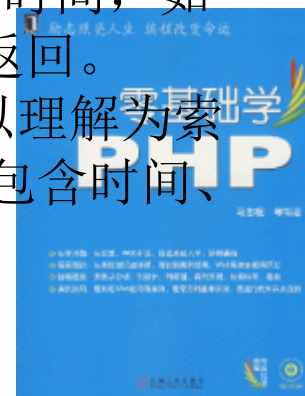
```
chmod(“/home/log/del.sh”, 0777);
```
- 函数chown(\$filename, \$user)，将文件\$filename的所有者改为\$user。如果更改成功，函数返回TRUE，否则函数返回FALSE。下面是该函数的一个示例代码。这行代码将文件/etc/passwd的所有者改为root。

```
chown(“/etc/passwd”, “root”);
```



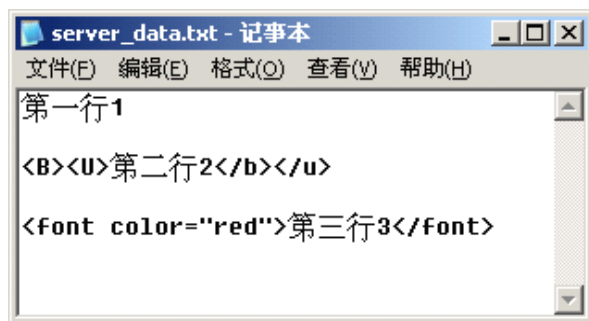
6.2.12 获取文件时间属性

- PHP提供了3个函数获取文件的时间属性，如文件的上次访问时间、文件的最后修改时间、取得文件的 inode 修改时间。这3个函数如下所示。
- 函数fileatime ()，返回文件上次被访问的时间，如果出错则返回FALSE。时间以UNIX时间戳的方式返回，关于 UNIX 时间戳的概念，读者可以先参见第 7.2 节的有关内容。
- 函数filemtime ()，返回文件上次被修改的时间，出错时返回FALSE。时间以UNIX时间戳的方式返回。
- 函数filectime ()，返回文件上次 inode 被修改的时间，如果出错则返回FALSE。时间以Unix时间戳的方式返回。inode是UNIX系统中的一个概念，其中文含义可以理解为索引节点，它它用来存放档案及目录的基本信息，包含时间、档名、使用者及群组等。

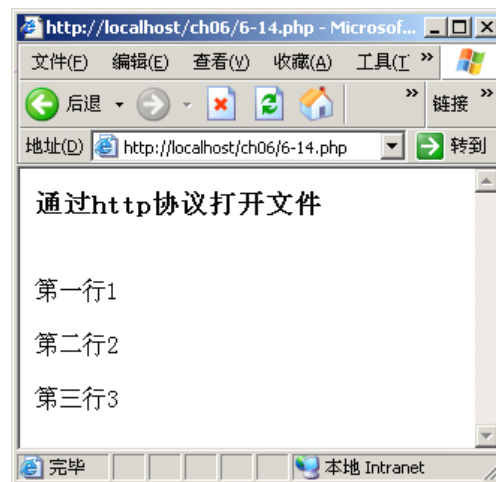


6.2.13 通过HTTP协议打开文件

- 通过HTTP协议打开文件，就是使用函数fopen()打开一个远程文件，这和使用函数fopen()打开一个本地文件几乎没有区别。下面通过一个具体示例程序说明如何通过HTTP协议打开文件，



服务器上某文件内容



通过HTTP协议打开文件后输出其内容



6.2.14 包含指定的文件到当前文件

- 最后这一小节要介绍两个比较重要的PHP语句——include和require。事实上，它们属于PHP流程控制方面的内容，之所以放在这里讲述，是因为它们涉及到文件和函数的概念。
- include()语句用来在当前文件中包含并运行指定文件。当一个文件被包含时，被包含文件中可用的任何变量、函数，在被调用的文件中也都可用，所有在包含文件中定义的变量、函数或类都具有全局作用域。下面举一个简单例子说明include的用法，



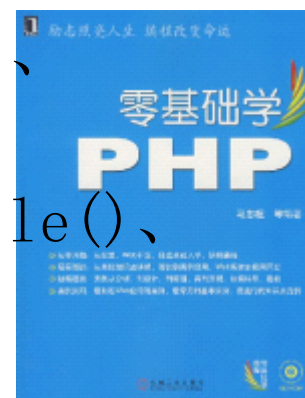
6.3 小结

- 本章讲述了如何使用PHP预定义函数，完成对文件系统的操作。首先介绍了和目录操作的有关函数，它们是：
- 打开目录函数opendir和关闭目录函数closedir()。
- 获取目录下文件名的函数readdir()。
- 列出某一目录下所有文件的函数scandir()。
- 获取和改变PHP工作目录的函数：getcwd()和chdir()。



6.3 小结

- 然后，介绍了和文件操作有关的函数，包括如下内容。
- 打开和关闭文件的函数：fopen()和fclose()
- 读取文件内容的函数：fgets()、readfile()、file()和file_get_contents()。
- 向文件写入内容的函数：fwrite()、fputs()和file_put_contents()。
- 获取文件有关信息的函数：fileowner()、filesize()和filetype()。
- 判断文件性质的函数：is_dir()、is_file()、is_readable()和is_writeable()。



6.3 小结

- 获取路径中文件名和目录名的函数：basename() 和dirname()。
- 判断文件或目录是否存在的函数：file_exists()。
- 建立和删除目录的函数：mkdir()和rmdir()。
- 拷贝、删除、移动文件的函数：copy()、unlink() 和rename()。
- 处理文件锁定的函数：flock()。
- 更改文件属性的函数：chgrp()、chmod()和chown()。
- 获取文件属性的函数fileatime()、filemtime() 和last_modify_inode()。

