File()函数：获取文件类型

实例： $fn2="/wamp";

echo '文件类型：'.filetype($fn2)."<br>";

file\_exists()函数：检查文件或目录是否存在。返回值为bool。

获取文件属性的函数

|  |  |
| --- | --- |
| Int filesize | 获取文件大小 |
| Int filectime | 获取文件的创建时间 |
| Int filemtime | 获取文件的修改时间 |
| Int fileatime | 获取文件的上次访问时间 |
| Bool is\_readable | 判断给定文件是否可读 |
| Bool is\_writable | 判断给定文件是否可写 |
| Bool is\_executable | 判断给定文件是否可执行 |
| Bool is\_file | 判断给定文件是否为一个正常的文件 |
| Bool is\_dir | 判断给定文件是否是一个目录 |
| Array stat | 给出文件的信息 |

$fn="../web/1.php"; //绝对路径C:\wamp\www\web/1.php

if(file\_exists($fn)&is\_file($fn))

{

echo '文件大小：'.filesize($fn).'字节<br>';

echo '文件的创建时间为'.date("Y-m-d H:i:s",filectime($fn))."<br>";

echo is\_readable($fn)?'该文件可读<br>':"该文件不可读<br>";

}

else

echo "not";

$fn="../web/1.php"; //绝对路径C:\wamp\www\web/1.php

if(file\_exists($fn)&is\_file($fn))

{

echo "<pre>";

print\_r(stat($fn));

echo "</pre>";

}

else

echo "not";

fopen()函数：打开文件

resource fopen( string $filename, string $mode[ ,bool $use\_include\_path=false [,resource $context] ] )

$filename表示指定打开的文件，$mode表示打开的模式，可选参数$use\_include\_path表示是否需要在include\_path中搜寻文件，可选参数$context表示上下文，用来设置一些其他的协议。

文件打开模式

|  |  |
| --- | --- |
| r | 读写方式打开，将文件指针指向文件头。 |
| r+ | 读写方式打开，将文件指针指向文件头。 |
| w | 写入方式打开，将文件指针指向文件头并将文件大小截为0，如果文件不存在尝试创建之。 |
| w+ | 读写方式打开，将文件指针指向文件头并将文件大小截为0，如果文件不存在尝试创建之。 |
| a | 写入方式打开，将文件指针指向文件末尾，如果文件不存在尝试创建之。 |
| a+ | 读写方式打开，将文件指针指向文件末尾，如果文件不存在尝试创建之。 |

fclose()函数：关闭文件

Bool fclose( resource $handle)

fread()函数：读取指定长度的字符串

string fread( resource $handle , int $length)

$fn="../1.txt"; //绝对路径C:\wamp\www\web/1.php

$handle1=fopen($fn,"r");

$content1=fread($handle1,9);

echo $content1."<br>";

fclose($handle1);

$handle2=fopen($fn,"r");

$content2=fread($handle2,filesize($fn));

echo $content2;

fclose($handle2);

feof：测试文件指针是否到达文件结束的位置

bool feof( resource $handle)

<?php

$fn=fopen("../1.txt","r");

while(!feof($fn)) //文件指针不指到末尾不停止循环

{

echo fgets($fn)."<br>";

}

fclose($fn);

?>

fgetc()函数：用于在打开的文件中读取一个字符。

string fgetc( resource $handle)

fgets()函数：用于在打开的文件中读取一行。

string fgetc( int $handle [, int $length])

$length指定了一行读取的字节数，碰到换行符、EOF或者已经读取了length-1字节后停止，如果没有指定$length，则默认为1024字节。

file\_get\_contents()函数：将文件的全部内容读取到一个字符串中。

string file\_get\_contents (string $filename[, bool $use\_include\_path[, resource $context[, int $offset[, int $maxlen ]]]])

$fn="../1.txt";

$handle1=file\_get\_contents($fn);

echo $handle1;

此函数是用来将文件内容读取到一个字符串中的首选方法。

file()函数：将整个文件读入到数组中，如果该函数执行成功，则返回一个数组，数组中的每个元素都是文件中的一行，包括换行符，失败返回false。

array file(string $filename[, int $use\_include\_path[, resource $context]])

$fielname指定了读取的文件名，也就是说该函数不用fopen()打开文件。

$lines=file("http://www.itcast.cn/");

foreach($lines as $line\_num => $line)

{

echo "Line #<b>{$line\_num}</b>:".htmlspecialchars($line)."<br>\n";

}

fwirte()函数：用于写入文件。

int fwrite( resource $handle, string $string[, int $length])

$length如果省略，则写入整个字符串

$filename="../1.txt";

$content="似乎不关闭文件就不能进行读取?";

$handle=fopen($filename,"r");

fwrite($handle,$content);

fclose($handle);

$handle2=fopen($filename,"a+");

$content2=fread($handle2,filesize($filename));

echo "succes!!!";

echo $content2;

似乎不关闭文件就不能进行读取。。。

file\_put\_contents()函数：对文件进行写入操作，且不需要使用fopen()。

int file\_put\_contents( string $filename, mixed $data,[, int $flags[, resource $context ] ] )

知道有这个函数就好。

**文件加锁机制**

并发访问：指多个浏览器在同一时刻访问服务器上的某一个文件，这意味着不同的访问进程会在同一时刻读写同一文件，很有可能造成数据的紊乱或者文件的损坏，为了避免这个问题，php提供了文件加锁机制。

flock()函数：文件加锁

bool flock( resource $handle, int $operation,[, int, $wouldblock ])

$handle参数表示文件系统指针，$operation指定了使用哪种锁类型，

$wouldblock为可选，若设置为1或true，则当进行锁定时阻挡其他进程。

$operation参数

|  |  |
| --- | --- |
| LOCK\_SH: | 取得共享锁定(读文件时使用) |
| LOCK\_EX: | 取得独占锁定(写文件时使用) |
| LOCK\_UN: | 释放锁定(无论共享或独占) |
| LOCK\_NB: | 不希望flock()在锁定时堵塞，则加上 |

copy()函数：实现文件拷贝功能

bool copy (string $source, string $dest)

参数$source表示源文件，参数$dest表示指定目标文件。

rename()函数：实现文件或目录的重命名功能

bool rename(string $oldname, string $newname[, resource $context])

$oldname表示指定源文件的名称，$newname表示指定新的文件名称

注意：如果两个文件在同一目录下，则是重命名操作，在不同目录下，则为移动操作。

**1.文件上传表单**

表单提交方式必须为POST，

还需要添加上传的属性enctype=”multipart/form-data”,

说明浏览器可以提供文件上传功能，服务器端提交的数据中包含文件的数据。