1. 文件检索

echo \_\_FILE\_\_ ; // 取得当前文件的绝对地址，结果：D:\www\test.php

echo dirname(\_\_FILE\_\_); // 取得当前文件所在的绝对目录，结果：D:\www\

echo dirname(dirname(\_\_FILE\_\_)); //取得当前文件的上一层目录名，结果：D:\

?>

1. php常用文件操作函数
   1. basename()返回路径中的文件名。分为带扩展名和不带扩展名的。
   2. copy() 复制文件。
   3. dirname() 返回路径中的目录部分。
   4. disk\_free\_space() 返回目录的可用空间。
   5. disk\_total\_space() 返回一个目录的磁盘总容量。
   6. file\_exists() 检查文件或目录是否存在。
   7. file\_get\_contents() 将文件读入字符串。
   8. file\_put\_contents() 将字符串写入文件。
   9. filesize() 返回文件大小。
   10. filetype() 返回文件类型。
   11. glob() 返回一个包含匹配指定模式的文件名/目录的数组。
   12. is\_dir() 判断指定的文件名是否是一个目录。
   13. is\_writable() 判断文件是否可写。
   14. mkdir() 创建目录
   15. move\_uploaded\_file() 将上传的文件移动到新位置。
   16. parse\_ini\_file() 解析一个配置文件。
   17. realpath() 返回绝对路径名。
   18. rename() 重名名文件或目录。
   19. tempnam() 创建唯一的临时文件。
   20. tmpfile() 建立临时文件。
   21. unlink() 删除文件。
   22. chmod() 改变文件模式。
   23. chown() 改变文件所有者。
   24. chgrp() 改变文件组。
2. php重定向的三种方法
3. 用HTTP头信息

<?php  
$url = "//www.jb51.net";  
if (isset($url))  
{  
Header("Location: $url");  
}  
?>

1. 用HTML标记

<?php  
$url = "//www.jb51.net";  
if (!isset($url))  
{  
exit('没有要跳转的地址呦');  
}  
?>  
<HTML>  
<HEAD>  
<META HTTP-EQUIV="REFRESH" CONTENT="5; URL='<?php echo $url;?>' ">  
</HEAD>  
<BODY>  
</BODY>  
</HTML>

1. 用脚本来实现

<? php  
$url="//www.jb51.net";  
echo "<SCRIPT LANGUAGE=\"JavaScript\">location.href='$url'</SCRIPT>";  
?>

1. 数据提交

<form id="form">

<input type="text" name="username"/>

<input type="text" name="password"/>

<button type="button" id="login">登录</button>

</form>

<script>

$('#login').click(function () {

$.post("{:url('')}", $('#form').serialize(),function (data) {

console.log(data);

})

})

</script>

1. php按行读取文件内容

$file = fopen("username.txt", "r");

$user=array();

$i=0;

//输出文本中所有的行，直到文件结束为止。

while(! feof($file))

{

 $user[$i]= fgets($file);//fgets()函数从文件指针中读取一行

 $i++;

}

fclose($file);

$user=array\_filter($user);

print\_r($user);

1. php常用数据操作函数

**mysql\_affected\_rows： 得到 MySQL 最后操作影响的列数目。**  
**mysql\_close： 关闭 MySQL 服务器连接。**  
**mysql\_connect： 打开 MySQL 服务器连接。**  
**mysql\_create\_db： 建立一个 MySQL 新数据库。**  
**mysql\_data\_seek： 移动内部返回指针。**  
**mysql\_db\_query： 送查询字符串 (query) 到 MySQL 数据库。**  
**mysql\_drop\_db： 移除数据库。**  
**mysql\_errno： 返回错误信息代码。**  
**mysql\_error： 返回错误信息。**  
**mysql\_fetch\_array： 返回数组资料。**  
**mysql\_fetch\_field： 取得字段信息。**  
**mysql\_fetch\_lengths： 返回单列各栏资料最大长度。**  
**mysql\_fetch\_object： 返回类资料。**  
**mysql\_fetch\_row： 返回单列的各字段。**  
**mysql\_field\_name： 返回指定字段的名称。**  
**mysql\_field\_seek： 配置指针到返回值的某字段。**  
**mysql\_field\_table： 获得目前字段的资料表 (table) 名称。**  
**mysql\_field\_type： 获得目前字段的类型。**  
**mysql\_field\_flags： 获得目前字段的标志。**  
**mysql\_field\_len： 获得目前字段的长度。**  
**mysql\_free\_result： 释放返回占用内存。**  
**mysql\_insert\_id： 返回最后一次使用 INSERT 指令的 ID。**  
**mysql\_list\_fields： 列出指定资料表的字段 (field)。**  
**mysql\_list\_dbs： 列出 MySQL 服务器可用的数据库 (database)。**  
**mysql\_list\_tables： 列出指定数据库的资料表 (table)。**  
**mysql\_num\_fields： 取得返回字段的数目。**  
**mysql\_num\_rows： 取得返回列的数目。**  
**mysql\_pconnect： 打开 MySQL 服务器持续连接。**  
**mysql\_query： 送出一个 query 字符串。**  
**mysql\_result： 取得查询 (query) 的结果。**  
**mysql\_select\_db： 选择一个数据库。**  
**mysql\_tablename： 取得资料表名称。**

1. Php获取当前系统时间

**date('Y-m-d h:i:s', time())**

1. 查看网页cookie

按F12进入浏览器的开发者模式——console——在命令行输入javascript:alert(document.cookie)，再回车

1. PHP简单获取及判断提交来源的方法

$\_SERVER['HTTP\_REFERER']; 获取上个页面的url

strrpos() - 查找字符串在另一字符串中最后一次出现的位置（区分大小写）