swoole:

<https://wiki.swoole.com/wiki/page/478.html>

1. PHP Loosely typed language

2. require() 函数与 include() 相同，不同的是它对错误的处理方式。include() 函数会生成一个警告（但是脚本会继续执行），而 require() 函数会生成一个致命错误（fatal error）（在错误发生后脚本会停止执行）。

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

while(!feof($file))

{

echo fgets($file); //逐行读取

echo fgetc($file); //逐字符读取

}

fclose($file);

4. $\_FILES["file"] 上传文件

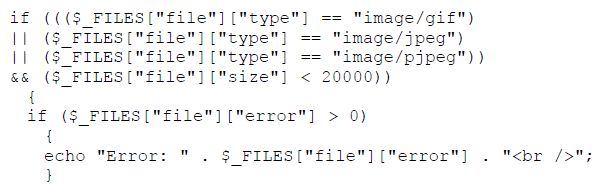
$\_FILES["file"]["name"] - 被上传文件的名称

$\_FILES["file"]["name"] - 被上传文件的名称

$\_FILES["file"]["size"] - 被上传文件的大小，以字节计

$\_FILES["file"]["tmp\_name"] - 存储在服务器的文件的临时副本的名称

$\_FILES["file"]["error"] - 由文件上传导致的错误代码



5. cookie session

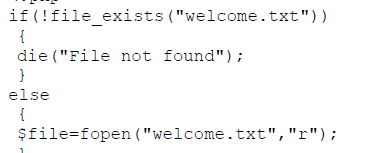
删除某些 session 数据，可以使用 unset() 或 session\_destroy() 函数

6. PHP错误处理

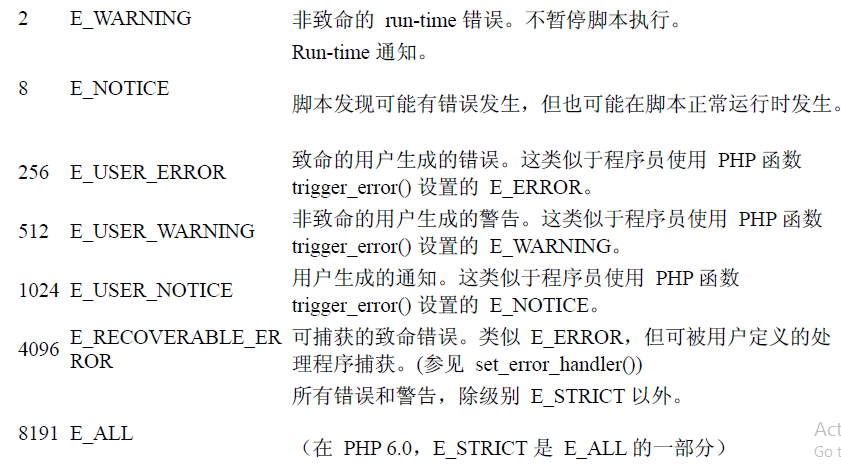
简单的 "die()" 语句

自定义错误和错误触发器

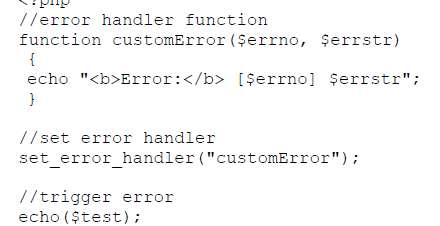
错误报告



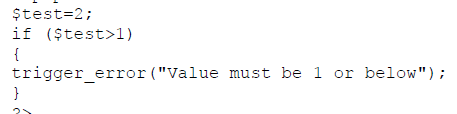
error\_function(error\_level,error\_message,error\_file,error\_line,error\_context)

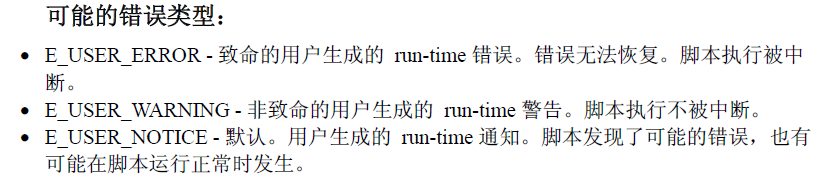


Set Error Handler：



触发错误trigger\_error()：





7. PHP 异常处理

当异常被抛出时，其后的代码不会继续执行，PHP 会尝试查找匹配的 "catch" 代码块。 如果异常没有被捕获，而且又没用使用 set\_exception\_handler() 作相应的处理的话，那么将发生一个严重的错误（致命错误），并且输出 "Uncaught Exception" （未捕获异常）的错误消息。

set\_exception\_handler() 函数可设置处理所有未捕获异常的用户定义函数。

需要进行异常处理的代码应该放入 try 代码块内，以便捕获潜在的异常。

每个 try 或 throw 代码块必须至少拥有一个对应的 catch 代码块。

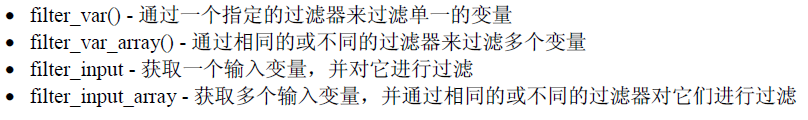
使用多个 catch 代码块可以捕获不同种类的异常。

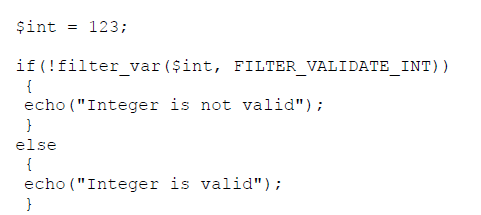
可以在 try 代码块内的 catch 代码块中再次抛出（re-thrown）异常。

简而言之：如果抛出了异常，就必须捕获它。

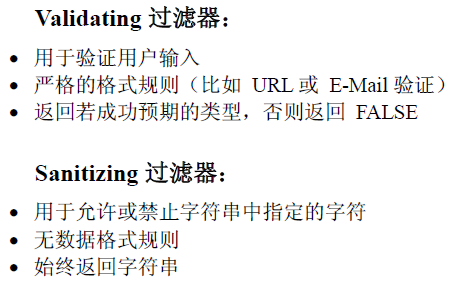
8. PHP 过滤器（Filter）

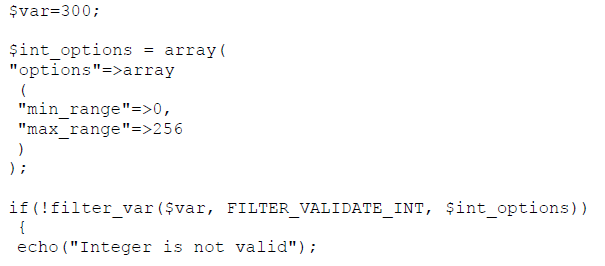
PHP 过滤器用于验证和过滤来自非安全来源的数据，比如用户的输入。



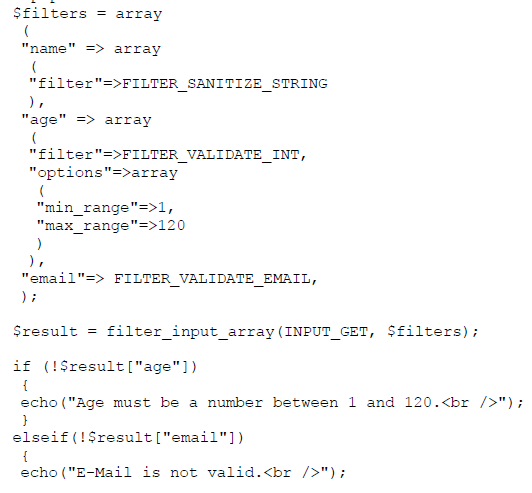


Validating 和 Sanitizing

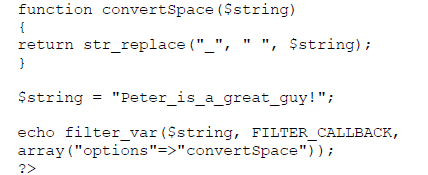




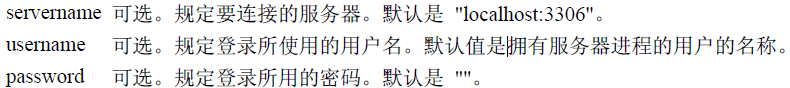
过滤多个输入：

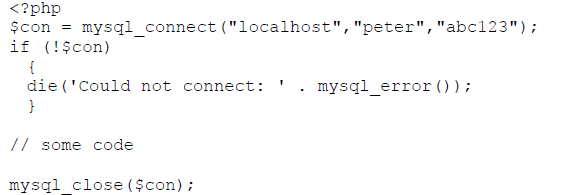


使用 Filter Callback



9. mysql\_connect(servername,username,password);

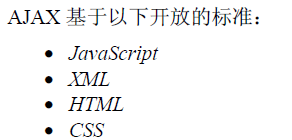




10. json\_encode()

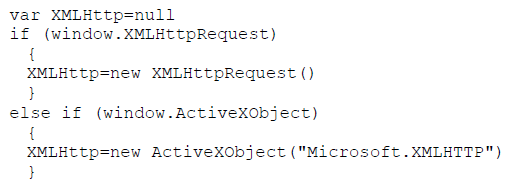
11. Ajax Asynchronous JavaScript And XML

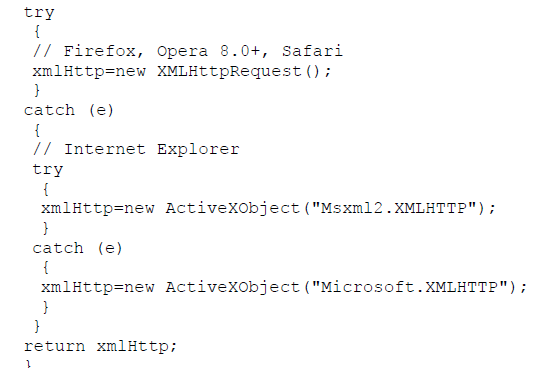
AJAX 并不是一种新的编程语言，而仅仅是一种新的技术，它可以创建更好、更快且交互性更强的 web 应用程序。AJAX 使用 JavaScript 在 web 浏览器与 web 服务器之间来发送和接收数据。通过在幕后与 web 服务器交换数据，而不是每当用户作出改变时重载整个 web 页面，AJAX 技术可以使网页更迅速地响应。



不存在什么 AJAX 服务器。AJAX 是一种在浏览器运行的技术。它使用浏览器与 web 服务器之间的异步数据传输，使网页从服务器请求少量的信息，而不是整张页面。

XMLHttpRequest:



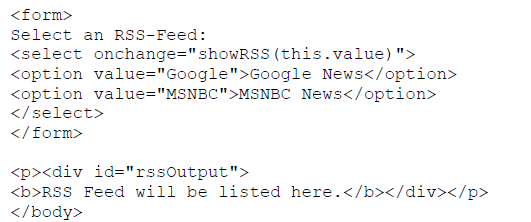


12. live search:

clipboard.png

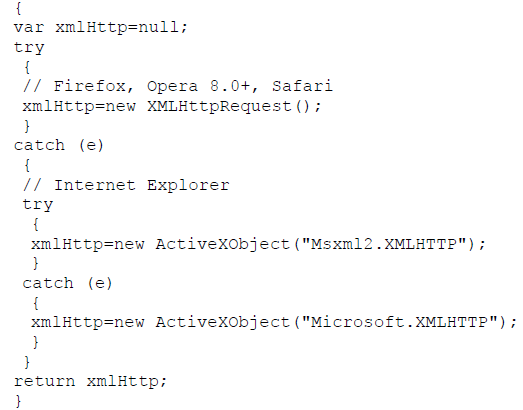
13. Ajax RSS

我们将演示一个 RSS 阅读器，通过它，来自 RSS 的内容在不进行刷新的情况下载入网页



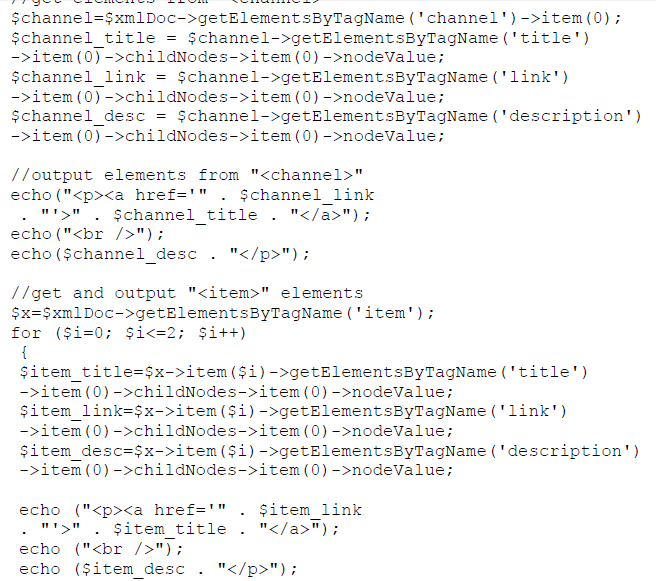
js:



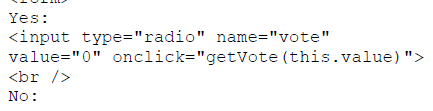


php:



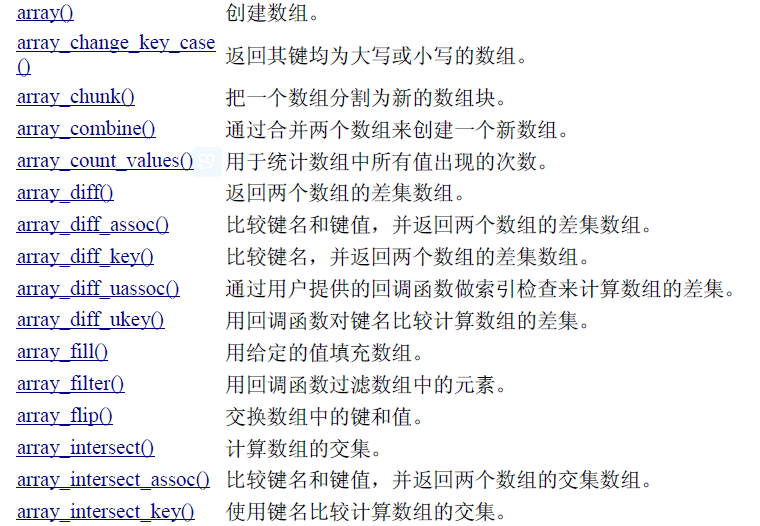


投票：

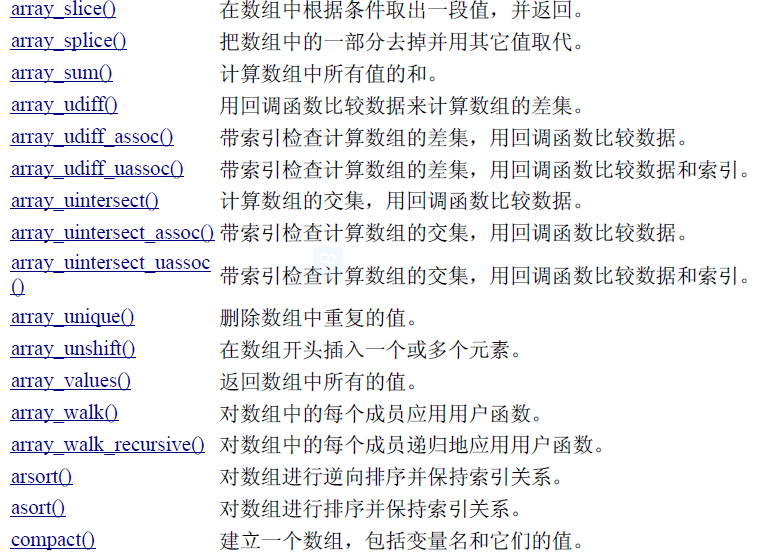


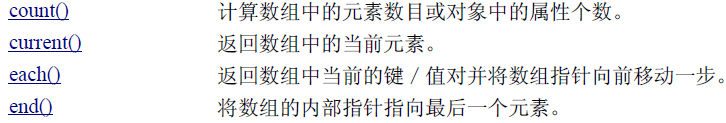
clipboard.png

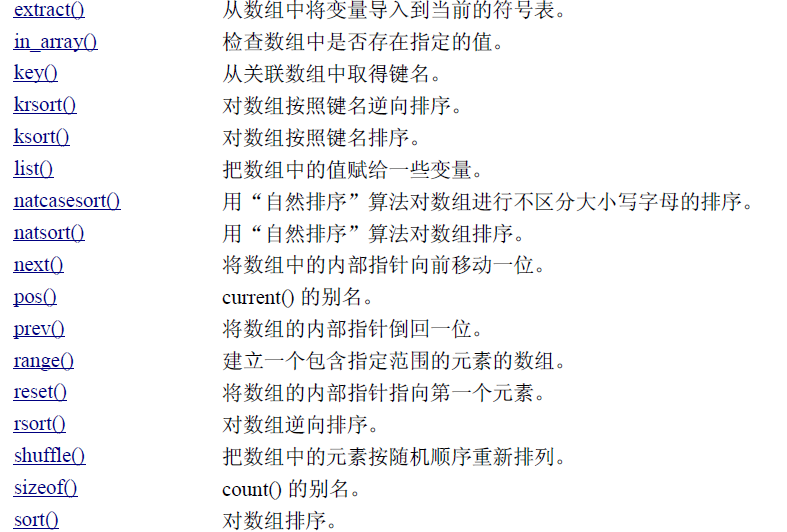
14. PHP Array

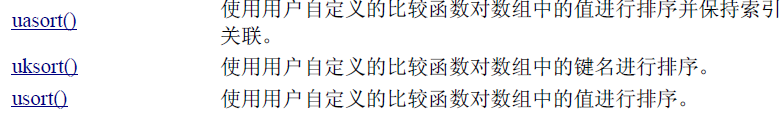




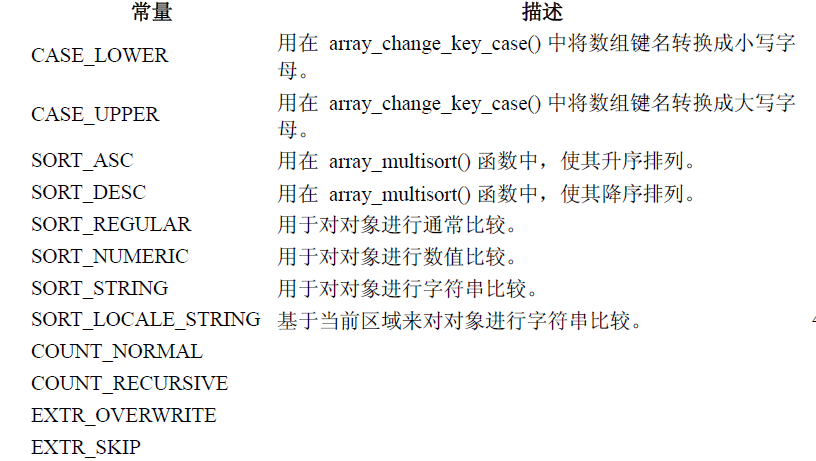


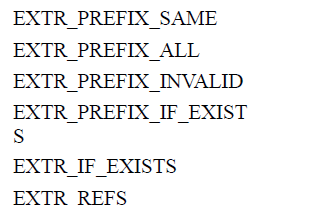




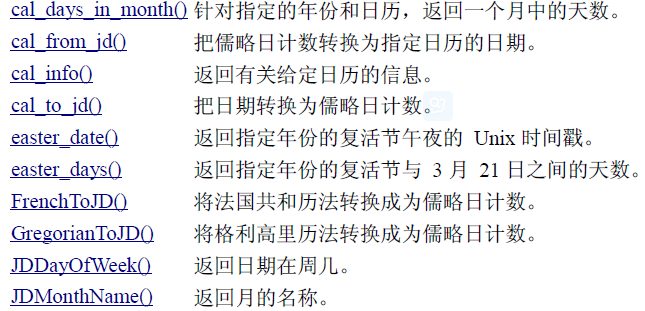


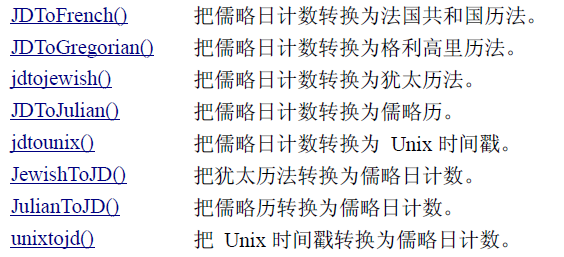
PHP Array 常量：





15. PHP Calendar 函数



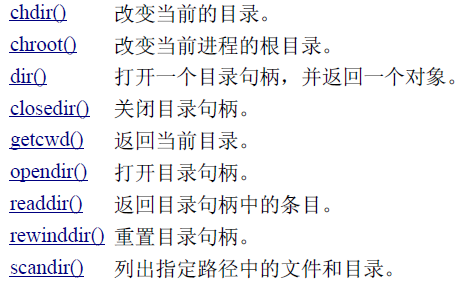


PHP Date / Time 函数：

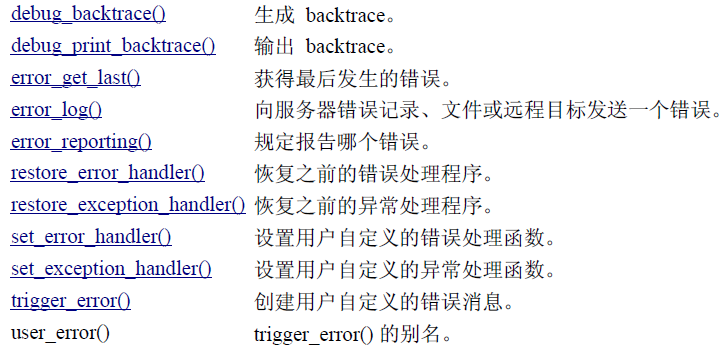
clipboard.png

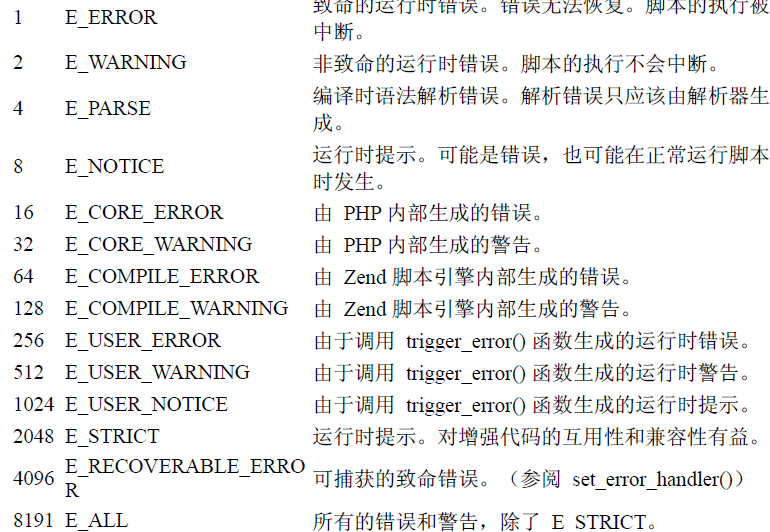


16. PHP Directory 函数

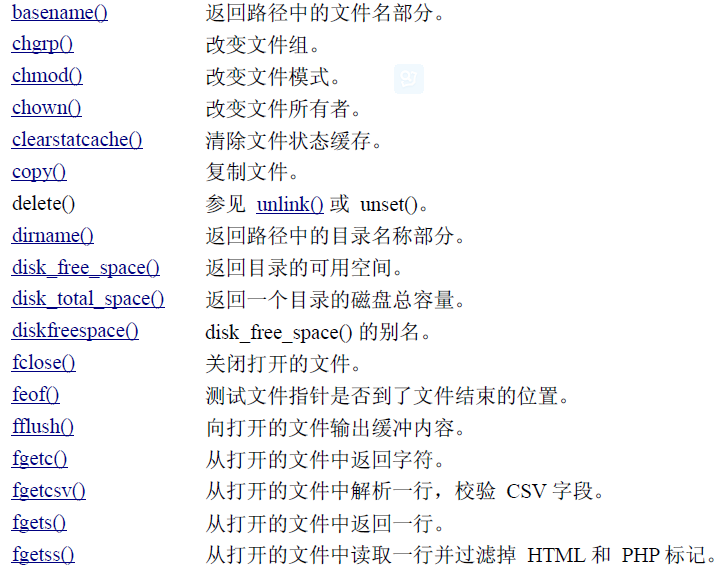


17. PHP Error 和 Logging 函数

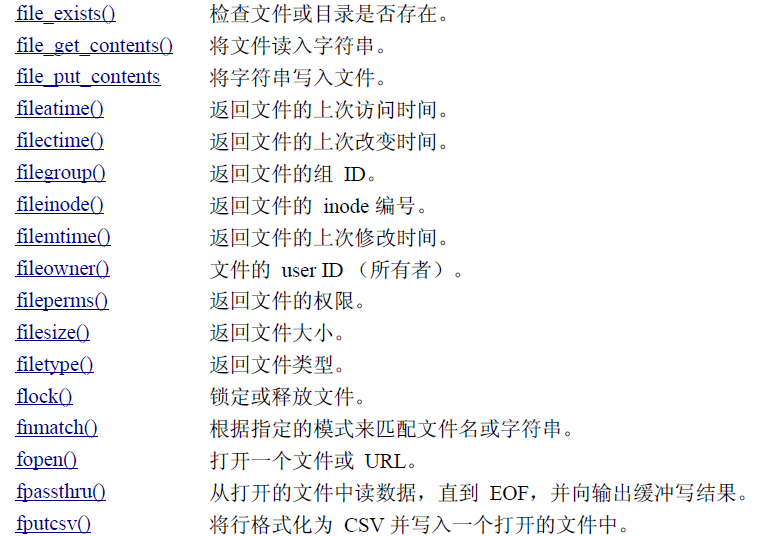




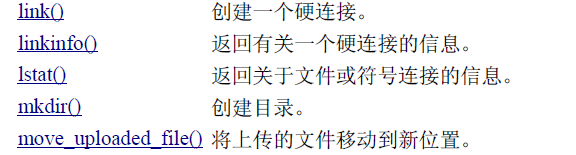
18. PHP Filesystem 函数



clipboard.png

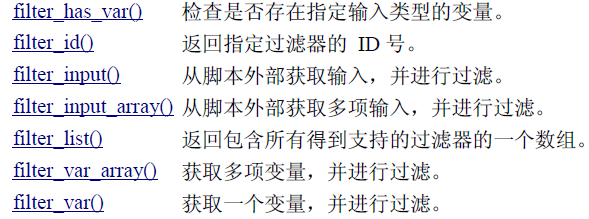




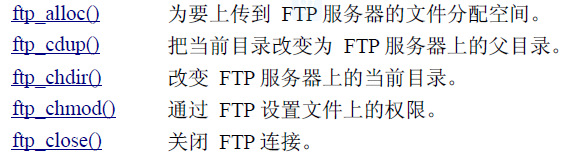


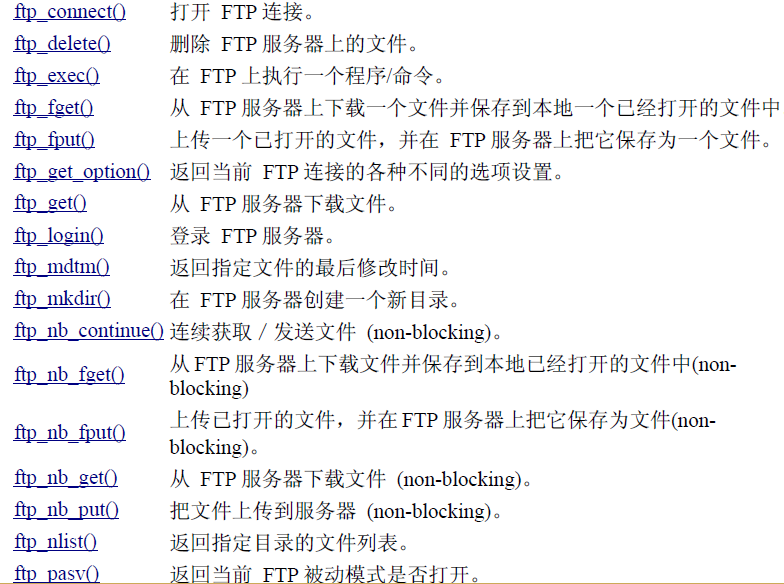


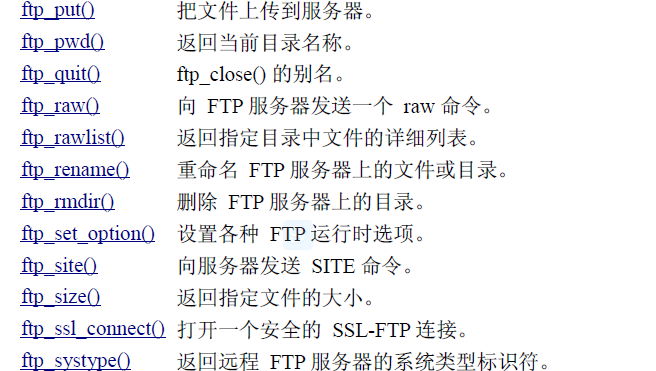
19. PHP Filter 函数



20. PHP FTP 函数







PHP HTTP 函数：

