

# php-1.19

## php错误处理

在php中，默认的错误处理很简单，一条消息会发送到浏览器，这条消息带有文件名、行号以及一条描述错误的消息。  
在创建脚本和web应用程序时，错误处理也是一个重要的部分，如果代码缺少错误检测编码，会增加安全风险

## 创建自定义错误处理器

创建一个自定义错误处理器，可以在php中发生错误时调用该函数  
该函数必须有能处理至少两个参数 (error level 和 error message)，但是可以接受最多五个参数（可选的：file, line-number 以及 error context）：

参数	描述
error_level	必需。为用户定义的错误规定错误报告级别。必须是一个值数。 参见下面的表格：错误报告级别。
error_message	必需。为用户定义的错误规定错误消息。

---

值	常量	描述
2	E_WARNING	非致命的 run-time 错误。不暂停脚本执行。 Run-time 通知。
8	E_NOTICE	脚本发现可能有错误发生，但也可能在脚本正常运行时发生。
256	E_USER_ERROR	致命的用户生成的错误。这类似于程序员使用 PHP 函数 trigger_error() 设置的 E_ERROR。
512	E_USER_WARNING	非致命的用户生成的警告。这类似于程序员使用 PHP 函数 trigger_error() 设置的 E_WARNING。
1024	E_USER_NOTICE	用户生成的通知。这类似于程序员使用 PHP 函数 trigger_error() 设置的 E_NOTICE。
4096	E_RECOVERABLE_ERROR	可捕获的致命错误。类似 E_ERROR，但可被用户定义的处理程序捕获。(参见 set_error_handler()) 所有错误和警告，除级别 E_STRICT 以外。
8191	E_ALL	(在 PHP 6.0, E_STRICT 是 E_ALL 的一部分)

## php异常处理

异常是用于在指定的错误发生时改变脚本的正常流程  
php5提供了一种新的面向对象的错误处理方法，异常处理用于在指定的错误情况发生时改变脚本的正常流程，这种情况称为异常。  
当异常被触发时，会发生：

- 当前代码状态被保持
- 代码执行被切换到预定义的异常处理器函数
- 根据情况，处理器也许会从保持的代码状态重新开始执行代码，终止脚本执行，或从代码中另外的位置继续执行脚本。

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。

处理处理程序应当包括：

- Try - 使用异常的函数应该位于 "try" 代码块内。如果没有触发异常，则代码将照常继续执行。但是如果异常被触发，会抛出一个异常。
- Throw - 这里规定如何触发异常。每一个 "throw" 必须对应至少一个 "catch"
- Catch - "catch" 代码块会捕获异常，并创建一个包含异常信息的对象