# 编码规范

## 一．开始标记和结束标记

当脚本中带有 PHP 代码，可以使用<?php ?>、<? ?>或<% %>等标记来界定 PHP 代码，

在 HTML 页面中嵌入纯变量时，还可以使用<?=$variablename ?>这样的形式。为了防止短标

记<? ?>和 ASP 风格的<% %>与一些技术发生冲突，有时需要在 PHP 配置文件中将其关闭，

因而导致这样的标记不总是可用。所以在编写 PHP 脚本时不允许使用短标记，所有脚本全

部使用完整的、标准的、PHP 定界标签<?php ?>作为 PHP 开始和结束标记。

对于只包含有 PHP 代码的文件，结束标志（"?>"）是不允许存在的，PHP 自身不需要

（"?>"）。 这样做, 可以防止它的末尾被意外地注入，从而导致像使用 Header()、setCookie()

和 session\_start()等设置头信息的函数时发生失败。

## 注释规范

注释是对于那些容易忘记作用的代码添加简短的介绍性内容，可以在 PHP 脚本中使用

以‚/\* \*‛开始和以‚\*/‛作为结束的多行文档注释、还有普通多行注释‚/\* \*/‛，以及单行

注释符号‚//‛。所有文档块建议和 phpDocumentor 格式兼容。PHPDocumentor 是一个用 PHP

写的工具，对于有规范注释的 php 程序，它能够快速生成具有相互参照,索引等功能的 API

文档。可以通过在客户端浏览器上操作生成文档，文档可以转换为 PDF,HTML,CHM 几种形

式，非常的方便。PHPDocument 是从你的源代码的注释中生成文档，因此在给你的程序做

注释的过程，也就是你编制文档的过程。从这一点上讲，PHPDocumentor 促使你要养成良

好的编程习惯，尽量使用规范，清晰文字为你的程序做注释，同时多少也避免了事后编制文

档和文档的更新不同步的一些问题。 在 PHPDocumentor 中，注释分为文档性注释和非文档

性注释。 所谓文档性注释，是那些放在特定关键字前面的多行注释，特定关键字是指能够

被 PHPDocumentor 分析的关键字，例如 class，var 等。

## 使用单引号声明字符串

单引号不需要去解析变量，也不需要解析全部的转义字符，所以解析的速度快。因些在

绝大多数可以使用单引号的场合，禁止使用双引号。依据上述分析，可以或必须使用单引号

的情况如下（但不限于此）：

字符串为固定值，不包含‚\t‛等特殊转义字符

$html = ’<input type=”text” name=”username” value=”admin” />’;

当字符串是不包含变量的文字，应当用单引号来括起来

$var = ‘value’;

关联数组的下标数组中，如果下标不是整型，而是字符串类型，请务必用单引号将

下标括起，正确的写法为$array[‘key’]，而不是$array[key]，因为不正确的写法会使

PHP 解析器认为 key 是一个常量，进而先判断常量是否存在，不存在时才以‚key‛

作为下标带入表达式中，同时出发错误事件，产生一条 Notice 级错误。

$array[‘key’] ；

数据库 SQL 语句中，所有数据必须加单引号，无论数值还是字串，以避免可能的

注入漏洞和 SQL 错误。

UPDATE users SET name=’admin’, age=’22’, height=’178.5’ where id=’1’

## 命名原则

用驼峰命名法进行文件目录命名，杜绝拼音或中英混合的命名方式。就一般约定而言，文件、目录、类、函数、变量和常量的名字，应该让代码阅读者能够容易的知道这些代码的作用。命名的原则就是以最少的字母达到最容易理解的意义。常量命名应具有描述性，常量名字必须用大写。

## 类名

一个文件中声明一个类，文件名中必须包含类名字符串，这些不仅容易查找，也有

利于实现在程序中自动加载类。而且需要使用合适的权限限制外部访问级别。

1 类中成员属性的声明必须放到类的顶部，

2 类名包括字母字符，不允许使用数字和其他字符

3 如果类名包括多个单词，应使用驼峰式命名方式，每个单词的第一个字母必须大写，

不允许连续大写。