# 编码规范

## 1 目的

本规范希望通过制定一系列规范化PHP代码的规则，以减少在浏览不同作者的代码时，因代码风格的不同而造成不便。

☆好处：特别是在多人的开放团队中，编码标准能帮助确保代码的质量、减少Bug和容易维护。程序员之间可以方便的了解他人的代码，弄清程序状况，就像自己写的代码一样。另一方面可以避免新人自创一套风格，使代码可读性降低，并养成终生的习惯。

## 1.2 目标

能够遵守公共一致的编码标准对任何开发项目都很重要，特别是在多人的开发团队中， 编码标准能帮助确保代码的质量、减少 Bug 和容易维护。

* 适应环境，方便的融入到项目团队中
* 在一致的环境下，团队协作中有更高的效率，团队的成员可以减少犯错的机会
* 程序员可以方便的了解其他人的代码，弄清程序的状况，就和看自已的代码一样
* 防止自创一套风格并养成终生的习惯，一次次的犯同样的错误

## 1.3 PHP开始和结束标记

* 统一使用完整的、标准的定界标记（<?php ?>）
* 对于只有包含PHP代码的文件，不使用结束标记（?>）。防止被意外注入，导致header()、setcookie()、session\_start()等设置头信息的函数发生失败

## 1.4命名约定

* 类文件都是以“.class.php“为后缀，且类文件名只允许字母，使用驼峰法命名，并且首字母大写，例如：DbMysql.class.php 。
* 配置和函数等其他类库文件之外的文件一般是分别以“.inc.php“和”.php“为后缀，且文件名命名使用小写字母和下划线的方式，多个单词之间以下 划线分隔，例如config.inc.php ， common.php，install\_function.php 。
* 确保文件的命名和调用大小写一致，是由于在类Unix系统上面，对大小写是敏感的。
* 类名和文件名一致（包括上面说的大小写一致），且类名只允许字母，例如 UserAction类的文件命名是UserAction.class.php， InfoModel类的文件名是InfoModel.class.php 。
* 控制器类以Action为后缀，例如 UserAction、InfoAction ，模型类以Model为后缀，例如UserModel、InfoModel ，其他类也分别以相应分类为后缀，例如Service 、Widget。
* 方法名只允许由字母组成，下划线是不允许的，首字母要小写，其后每个单词首字母要大写，即所谓的 “驼峰法命名” 规则，且越详细越好，应该能够描述清楚该方法的功能，例如switchModel、findPage。
* 属性的命名只允许由字母组成，下划线是不允许的，首字母要小写，其后每个单词首字母要大写，即所谓的 “驼峰法命名” 规则，例如tablePrefix、tableName 。
* 对于对象成员的访问，我们必须始终使用 “get” 和 “set” 方法。例如：  
  class Foo  
  {  
  protected $\_testObj;  
  public function getTestObj()

{  
return $this->\_testObj;  
}  
public function setTestObj($testObj)

{  
$this->testObj = $\_testObj;  
}  
}

## 1.5 文件格式

* 缩进应该能够反映出代码的逻辑结果，尽量使用四个空格，禁止使用制表符TAB，因为这样能够保证有跨客户端编程器软件的灵活性。  
  例如：  
  if (1 == $x) {  
  $indented\_code = 1;  
  if (1 == $new\_line) {  
  $more\_indented\_code = 1;  
  }  
  }
* 变量赋值必须保持相等间距和排列。  
  例如：  
  $variable = 'demo';  
  $var = 'demo2';
* 每行代码长度应控制在80个字符以内，最长不超过120个字符。因为 linux 读入文件一般以80列为单位，就是说如果一行代码超过80个字符，那么系统将为此付出额外操作指令。这个虽然看起来是小问题，但是对于追求完美的程序员来说也是值得注意并遵守的规范。
* 每行结尾不允许有多余的空格。

## 1.6 注释规范

1. 所有的文档注释都是用/\*\*开始以\*/结束
2. 临时性代码注释都是用//debug开始的单行注释

## 1.7 命名原则

所有命名必须具有描述性，杜绝使用拼音，尽量不要使用数字

* 文件命名原则

所有文件名必须小写

普通文件：名+.php

类文件 ：名+.class.php

函数库 ：名+.func.php

配置文件：名+.inc.php

* 类命名原则

首字母大写，如果是多个单词组合，则所有单词首字母都大写

* 函数名或方法名

函数或方法名中必须要有描述性，且尽量带有动词描述（get、set等)

* 常量名

常量名必须使用大写，如果是多个单词使用下划线分开

* 全局变量

以 **$global\_** 开头

## 1.8 函数的定义与使用

* 声明函数时**参数的名字和变量的命名规范一致**；
* 函数定义中的左小括号，与函数名紧挨，中间无需空格；
* 开始的左大括号与函数定义为同一行，不要另起一行；
* 如果使用具有默认值的参数，应该位于参数列表的后面；
* 必须仔细检查并切实杜绝函数起始缩进位置与结束缩进位置不同的现象。

建议不要使用全局函数；