

PHP 编码规范

为什么需要编码规范？

- 为了提高工作效率，保证开发的有效性和合理性。
- 为了提高代码可读性和可重复利用性，从而节约沟通成本。

本文主要参考了 `PEAR` 规范，并进行适当的简化和调整。

主要介绍，命名规范、注释规范、代码风格。

文件标记

所有PHP文件，代码标记均使用完整的PHP标签，不建议使用短标签。

```
<?php
    echo 'Hello world';
?>

<?
//短标签不推荐
    echo 'Hello world';
?>
```

文件格式

文件编码 为无 BOM 的 UTF-8。

纯PHP类文件，文件最后 `?>` 要省略。

TextMate

在 "文件编码" 中，选择 "UTF-8（推荐）"

在 "换行符" 中，选择 "LF（推荐）"

文件命名

程序的文件名和目录名都采用有意义的英文命名。

不使用拼音或无意义的字母。

只允许出现字母、数字、下划线、中划线字符。

多个词之间使用驼峰命名法。

//类统一采用

demoTest.class.php

//接口统一采用

demoTest.interface.php

//其他按照各自的方式

demoTest.{style}.php

//其他文件参照

demoTest.inc.php

demo.lib.php

文件目录结构命名

因使用的框架不同，可根据实际情况考虑目录结构。

全局变量命名

```
$_GLOBAL['_startTime_']  
  
or  
  
$_GLOBAL['g_startTime_']
```

两边都有“_”，中间使用驼峰命名。

普通变量命名

数据类型	命名规范
字符串	\$strMyStr
数组	\$arrMyArray
对象	\$objMyObject
布尔值	\$flagMyFlag

采用驼峰命名，建议在变量前加上变量的类型作为前缀。

变量应该以名词为准，尽量避免使用常用关键字或存在模糊意义的单词。

私有变量，建议加上前缀"_"。

函数命名

函数名即要有意义，也要尽量缩写，一看就知道干什么。

建议单用动词或动词加形容词的格式命名。

私有方法，建议在加上前缀"_"。

```
//例如
private function _showMsg()
{
    //方法体
}
```

不建议下面这样的函数名：

```
public function getAdvertisementByCategoryIdAndPositionIdAndScheduleId()
{
    //方法体
}
```

可修改为：

```
public function getAd($categoryid, $positionid, $scheduleid)
{
    //方法体
}
```

习惯与约定

为了减少变量的长度，在不影响可读性的前提下，习惯对变量进行缩写。

全称	缩写
image	img

string	str
database	db
array	arr
count	cnt
message	msg
password	passwd 或 pwd
...	...

以上规范可用于，PHP代码、JavaScript代码、数据库表字段命名等。

文件注释

```
/**
 * 文件的简述
 *
 * PHP Version 6(PHP版本)
 *
 * @category 可以写部门(英文)
 * @package 可以写模块(英文)
 * @author test <test@company.com>
 * @time 2017/02/02 11:48
 * @copyright 2017 公司名称
 * @license 公司网址 license
 * @link test@qq.com(作者联系方式)
 */
```

类注释

```
/**
 * 类的简述
 *
 * @category 可以写部门(英文)
 * @package 可以写模块(英文)
 * @author test <test@company.com>
 * @license 公司网址 license
 * @link test@qq.com(作者联系方式)
 */
```

方法注释

```
/**
 * 方法的简述
 * @param array $myArray 参数解释
 * @param string $myString 参数解释
 * @return array(返回数据类型)
 */
```

代码注释

注释写在被注释代码的前面，而不是后面，但对于单行语句，注释可写在语句末尾。

对于大段注释，使用 `/* */` 进行注释。

注释不宜太多，大家能看的懂得行不必注释。

代码注释应该描述为什么，而不是做什么。

不要为了注释而注释。

标注的使用

IDE 支持一些特殊注释，可以列出整个项目中的特殊注释，方便后期维护和代码检查。

例如：

`//@fixMe` 表示需要修复项。

`//@todo` 表示需要完善的地方。

代码风格

尽量保证程序语句一行就是一句。

尽量不要使一行的代码过长，一般控制在80个字符之间。

如果一行代码太长，请使用类似“`.=`”的方式断行书写。

类、方法的做大括号需要独占一行。

其他控制语句等大括号和表达式同一行，并空格隔开。

```

class Demo
{
    public function index()
    {
        for ($i = 1, $i < 10, $i++) {

        }
    }

    public function test()
    {
        if ($expr1) {
            //if body

        } elseif ($expr2) {
            //elseif body

        } else {
            //else body

        }

        foreach ($data as $key => $value) {
            //foreach body
        }

        switch ($expr1) {
            case 0:
                echo '零';
                break;
            case 1:
                echo '一';
                break;
            default:
                echo 'null';
                break;
        }

        //尽量同等意义的变量等号对其
        $strName      = $arrUserInfo['name'];
        $strAge        = $arrUserInfo['age'];
        $strBirthday   = $arrUserInfo['birthday'];
        $strHobby       = $arrUserInfo['hobby'];
    }
}

```

调试代码

不要在你的提交的代码中包含调试代码，就算是注释掉了也不行。

像 `var_dump()`、`__printr()`、`die()` 和 `exit()` 这样的函数。

PHP错误

运行代码时不应该出现任何错误信息，并不是把警告和提示信息关掉来满足这一点。

例如，绝不要直接访问一个你没设置过的变量，你应该先使用 `isset()` 函数判断下。

最后

最后说的是，本规范不是强制，也不是标准。

“约定大于规范”，如果有的规范太死板，不适应您的团队，您可以不采用，按照您自己的规范即可。

推荐**PHP开发IDE**：`PHPStorm`。

推荐阅读：

[Mac PHPStorm 使用心得](#)

[PHP团队开发中遇到的那些坑，看我是如何解决的？](#)



