# PHP 编码规范

### 1.PSR-1 基本代码规范

本篇规范制定了代码基本元素的相关标准，以确保共享的PHP代码间具有较高程度的技术互通性。

关键词 “必须”("MUST")、“一定不可/一定不能”("MUST NOT")、“需要”("REQUIRED")、 “将会”("SHALL")、“不会”("SHALL NOT")、“应该”("SHOULD")、“不该”("SHOULD NOT")、 “推荐”("RECOMMENDED")、“可以”("MAY")和”可选“("OPTIONAL")

* 1. **概览**

PHP代码文件**必须**以<?php 或<?= 标签开始;

PHP 代码文件**必须**以不带BOM的UTF-8编码;

PHP 代码中**应该**只定义类、函数、常量等声明，或其他会产生**从属效应** 的操作（如：生成文件输出以及修改.ini配置文件等），二者只能选其一;

命名空间以及类**必须**符合PSR的自动加载规范：PSR-4中的一个;

类的命名**必须**遵循StudlyCaps 大写开头的驼峰命名规范;

类中的常量所有字母都**必须**大写，单词间用下划线分隔;

方法名称**必须**符合camelCase 式的小写开头驼峰命名规范;

* 1. **命名空间和类**

命名空间以及类的命名必须遵循PSR-4

根据规范，每个类都独立为一个文件，且命名空间至少有一个层次：顶级的组织名称

类的命名**必须**遵循 StulyCaps 大写开头的驼峰命名规范

PHP5.3及以后版本的代码使用正式的命名空间

例：

<?php

namespace Vendor\Model;

class Foo

{

｝

* 1. **类的常量 属性和方法**

类的常量中所有字母都**必须**大写，词间以下划线分隔

参照一下代码:

<?php

namespace Vendor\Model;

class Foo

{

const VERSION = ‘1.0’

const DATE\_APPROVED = ‘2012-06-01’;

}

类的属性命名**可以**遵循:

大写开头的驼峰式（$StudlyCaps）

小写开头的驼峰式（$camelCase）

下划线分隔式（$under\_score）

本规范不做强制要求，但无论遵循哪种命名方式，都 应该 在一定的范围内保持一致。这个范围可以是整个团队、整个包、整个类或整个方法。

类的方法名**必须**符合camelCase()式的小写开头驼峰命名规范

### 2. PSR-2编码风格规范

本篇规范是 [PSR-1] 基本代码规范的继承与扩展。

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。

当多名程序员在多个项目中合作时，就需要一个共同的编码规范， 而本文中的风格规范源自于多个不同项目代码风格的共同特性， 因此，本规范的价值在于我们都遵循这个编码风格，而不是在于它本身。

**2.1 概览**

代码必须遵循PSR-1中的编码规范

代码必须使用4个空格而不是tab键

每行的字符数应该软性保持在80个之内,理论上一定不可多于120个，但一定不可有硬性限制

每个namespace 命名空间声明语句和 use声明语句后面，必须插入一个空白行

类的开始花括号（｛）必须在函数声明后自成一行

方法的开始花括号（｛）必须写在函数声明后自成一行，结束花括号（｝）也必须写在函数主体后自成一行

类的属性和方法必须添加访问修饰符（private,protected 以及public）, abstract 以及final 必须声明在访问修饰符之前，而static必须声明在访问修饰符之后

控制结构的关键字后必须要有一个空格符，而调用方法或函数时则一定不可有

控制结构的开始花括号（｛）必须写在声明的同一行 而结束花括号（｝）必须写在主体后自成一行

控制结构的开始左括号和结束右括号前，都一定不可有空格符

**2.2 例子：**

以下例子程序简单地展示了以上大部分规范：

<?php

namespace Vendor\Package;

use FooInterface;

use BarClass as Bar;

use OtherVendor\OtherPackage\BazClass;

class Foo extends Bar implements FooInterface

{

public function sampleFunction($a, $b = null)

{

If ($a === $b) {

bar();

} elseif ($a > $b) {

$foo->bar($arg1);

} else {

BazClass::bar($arg2, $arg3);

}

}

final public static function bar()

{

//方法的内容

}

}

**2.3 通则**

2.3.1 文件

所有PHP文件必须以一个空白行作为结束

纯PHP代码文件必须省略最后的 ?> 结束标签

2.3.2 行

非空行后一定不可有多余的空格符

空行可以使得阅读代码更加方便以及有助于代码的分块

每行一定不可存在多于一条的语句

2.3.4 缩进

使用空格而不是tab缩进好处在于 避免在比较代码差异，打补丁，重阅代码以及注释时产生混淆。

**2.4 namespace以及use声明**

namespace 声明后必须插入空白行

所有 use 必须在 namespace 后声明

每条use 声明语句必须只有一个use 关键词

use 声明语句块后必须要有一个空白行

例如：

<?php

namespace Vendor\Package;

use FooClass;

use BarClass as Bar;

use OtherVendor\OtherPackage\BazClass;

//更多的PHP 代码在这里

**2.5 类 属性 方法**

2.5.1 扩展与继承

关键词 extends 和 implements 必须写在类名称的同一行

类开始的花括号必须独占一行 结束花括号也必须在类主体后独占一行

<?php

namespace Vendor\Package;

use FooClass;

use BarClass as Bar;

use OtherVendor\OtherPackage\BazClass;

class ClassName extends ParentClass implement \ArrayAccess, \Countable

{

// 这里是常量 属性 类方法

}

implements继承列表也可以分成多行，这样的话， 每个继承接口名称都必须分开独立成行, 包括第一个

<?php

namespace Vendor\Package;

use FooClass;

use BarClass as Bar;

use OtherVendor\OtherPackage\BazClass;

class className extends ParentClass implements

\ArrayAccess,

\Countable,

\Serializable

{

//这里面是常量 属性 类方法

}

2.5.2 属性

每个属性都必须添加访问修饰符

一定不可使用关键字 var 声明一个属性

每条语句一定不可定义超过一个属性

不该使用下划线作为前缀 来区分属性是protected 或 private

<?php

namespace Vendor\Package;

class ClassName

{

public $foo = null;

}

2.5.3 方法

所有方法都必须添加访问修饰符

不该使用下划线作为前缀 来区分方法是protected或private

方法名称后一定不可有空格符，其开始花括号必须独占一行，结束花括号也必须在方法主体后单独成一行 参数左括号和右括号前一定不可有空格

一个标准的方法声明可参照以下范例，留意其括号，逗号，空格以及花括号的位置。

<?php

namespace Vendor\Package;

class ClassName

{

public function fooBarBaz($arg1, &$arg2, $args = [])

{

//method body

}

}

2.5.4 方法的参数

参数列表中，每个逗号后面必须要有一个空格 而逗号前面一定不可有空格。

有默认值的参数 必须放到参数列表末尾

<?php

namespace Vendor\Package；

class ClassName

{

public function foo（$arg1, &$arg2, $arg3 = []）

{

//method body

}

}

参数列表可以分列成多行 这样 包括第一个参数在内的每个参数必须单独成行

拆分成多行的参数列表后，结束括号以及开始花括号必须写在同一行，中间用一个空格分开

<?php

namespace Vendor\Package;

class ClassName

{

public function aVeryLongMethodName(

ClassTypeHint $arg1,

&$arg2,

array $arg3 = []

) {

// 方法的内容

}

}

2.5.5 abstract final static

需要添加 abstract final static 声明时 必须写在访问修饰符前 而static则必须写在其后

<?php

namespace Vendor\Package;

abstract class ClassName

{

protected static $foo;

abstract protected function zim();

final public static function bar()

{

//method body

}

}

2.5.6 方法及函数调用

方法及函数调用时 方法名或函数名与参数左括号之间一定不可有空格， 参数右括号前也一定不可有空格 每个参数前一定不可有空格 但其后必须有一个空格。

<?php

bar（）;

$foo->bar($arg1);

Foo::bar($arg2, $arg3);

参数可以分列成多行 此时包括第一个参数在内的每个参数都必须单独成行

<?php

$foo->bar(

$longArgument,

$longrtArgument,

$muchLongerArgument

);

**2.6 控制结构**

控制结构的基本规范如下：

控制结构关键词后 必须 有一个空格。

左括号 ( 后 一定不可有空格。

右括号 ) 前也 一定不可有空格。

右括号 ) 与开始花括号 { 间 必须 有一个空格。

结构体主体 必须要有一次缩进。

结束花括号 } 必须 在结构体主体后单独成行。

每个结构体的主体都 必须 被包含在成对的花括号之中， 这能让结构体更加结构话，以及减少加入新行时，出错的可能性。

2.6.1 if elseif else

标准的if 结构如下代码所示 请留意括号 空格 花括号的位置注意else 和 elseif 都与前面结束花括号在同一行

<?php

if （$expr1）{

// if body

} elseif ($expr2) {

//elseif body

} else {

//else body;

}

2.6.2 switch 和 case

标准的 switch 结构如下代码所示，留意括号、空格以及花括号的位置。 case 语句必须相对switch进行一次缩进， 而break语句以及case 内的其它语句必须相对case 进行一次缩进。

如果存在非空的 case 直穿语句，主体里必须有类似 //no break 的注释

<?php

switch ($expr) {

case 0:

echo 'First case, with a break';

break;

case 1:

echo 'Second case, which falls through';

// no break

case 2:

case 3:

case 4:

echo 'Third case, return instead of break';

return;

default:

echo 'Default case';

break;

}

2.6.3 while 和 do while

一个规范的 while 语句应该如下所示，注意其「括号」、「空格」以及「花括号」的位置。

<?php

while ($expr) {

//structure body

}

标准的 do while 语句如下所示，同样的，注意其「括号」、「空格」以及「花括号」的位置。

<?php

do {

// structure body;

} while ($expr);

2.6.4 for

标准的 for 语句如下所示，注意其「括号」、「空格」以及「花括号」的位置。

<?php

for ($i = 0; $i < 10; $i++) {

// for body

}

2.6.5 foreach

标准的 foreach 语句如下所示，注意其「括号」、「空格」以及「花括号」的位置

<?php

foreach ($iterable as $key => $value) {

// foreach body

}

2.6.6 try,catch

标准的 try catch 语句如下所示，注意其「括号」、「空格」以及「花括号」的位置

<?php

try {

// try body

} catch (FirstExceptionType $e) {

// catch body

} catch (OtherExceptionType $e) {

// catch body

}

2.7 闭包

闭包声明时，关键词 function 后以及关键词 use 的前后都 必须要有一个空格。

开始花括号必须写在声明的同一行，结束花括号必须紧跟主体结束的下一行。

参数列表和变量列表的左括号后以及右括号前，一定不可 有空格。

参数和变量列表中，逗号前 一定不可 有空格，而逗号后 必须 要有空格。

闭包中有默认值的参数 必须 放到列表的后面。

标准的闭包声明语句如下所示，注意其「括号」、「空格」以及「花括号」的位置。

<?php

$closureWithArgs = function ($arg1, $arg2) {

// body

};

$closureWithArgsAndVars = function ($arg1, $arg2) use ($var1, $var2)

｛

// body

}

**3.注释**

**PhpStrom 有很好的注释文档生成能力**

**PHPDoc 是PHP文档注释的标准，可以帮助我们在注释文档时有规范 可参见以下网址**

http://www.cnblogs.com/picaso/archive/2012/10/04/2711435.html