**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 gbk 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-0](https://github.com/PizzaLiu/PHP-FIG/blob/master/PSR-0-cn.md) 或 [PSR-4](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-4-autoloader.md) 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。

## 2. 文件

### 2.1. PHP标签

PHP代码**必须**使用 <?php ?> 长标签 或 <?= ?> 短输出标签；**一定不可**使用其它自定义标签。

### 2.2. 字符编码

PHP代码**必须**且只可使用不带BOM的gbk编码。

命名空间以及类的命名必须遵循 [PSR-0](https://github.com/PizzaLiu/PHP-FIG/blob/master/PSR-0-cn.md).

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（vendor name）。

类的命名必须 遵循 StudlyCaps 大写开头的驼峰命名规范。

PHP 5.3及以后版本的代码**必须**使用正式的命名空间

## 3. 类的常量、属性和方法

此处的“类”指代所有的类、接口以及可复用代码块（traits）

### 3.1. 常量

类的常量中所有字母都**必须**大写，词间以下划线分隔。

### 3.2. 属性

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) 又或者是 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。

### 3.3. 方法

方法名称**必须**符合 camelCase() 式的小写开头驼峰命名规范。

## 4. 通则

### 4.1 基本编码准则

代码**必须**符合 [PSR-1](https://github.com/PizzaLiu/PHP-FIG/blob/master/PSR-1-basic-coding-standard-cn.md) 中的所有规范。

### 4.2 文件

所有PHP文件**必须**使用Unix LF (linefeed)作为行的结束符。

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。

### 4.3. 行

行的长度**一定不能**有硬性的约束。

软性的长度约束**一定**要限制在120个字符以内，若超过此长度，带代码规范检查的编辑器**一定**要发出警告，不过**一定不可**发出错误提示。

每行**不应该**多于80个字符，大于80字符的行**应该**折成多行。

非空行后**一定不能**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**一定不能**存在多于一条语句。

### 4.4. 缩进

代码**必须**使用4个空格符的缩进，**一定不能**用 tab键 。

备注: 使用空格而不是tab键缩进的好处在于， 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。 并且，使用空格缩进，让对齐变得更方便。

### 4.5. 关键字 以及 True/False/Null

PHP所有 [关键字](http://php.net/manual/en/reserved.keywords.php)**必须**全部小写。

常量 true 、false 和 null 也**必须**全部小写。

## .4.6. namespace 以及 use 声明

namespace 声明后 必须 插入一个空白行。

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;

// ... additional PHP code ...

## 5. 类、属性和方法

此处的“类”泛指所有的class类、接口以及traits可复用代码块。

### 5.1. 扩展与继承

关键词 extends 和 implements**必须**写在类名称的同一行。

类的开始花括号**必须**独占一行，结束花括号也**必须**在类主体后独占一行。

implements 的继承列表也**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。

### 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。

**不要**使用下划线作为前缀，来区分属性是 protected 或 private。

### 5.3. 方法

所有方法都**必须**添加访问修饰符。

**不要**使用下划线作为前缀，来区分方法是 protected 或 private。

方法名称后**一定不能**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。参数左括号后和右括号前**一定不能**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。

### 5.4. 方法的参数

参数列表中，每个参数后面**必须**要有一个空格，而前面**一定不能**有空格。

有默认值的参数，**必须**放到参数列表的末尾。

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都**必须**单独成行。

拆分成多行的参数列表后，结束括号以及方法开始花括号 必须 写在同一行，中间用一个空格分隔。

### 5.5. abstract 、 final 、 以及 static

需要添加 abstract 或 final 声明时， **必须**写在访问修饰符前，而 static 则**必须**写在其后。

### 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**一定不能**有空格，参数右括号前也 **一定不能**有空格。每个参数前**一定不能**有空格，但其后**必须**有一个空格。

## 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**一定不能**有空格。
* 右括号 ) 前也**一定不**能有空格。
* 右括号 ) 与开始花括号 { 间**一定**有一个空格。
* 结构体主体**一定**要有一次缩进。
* 结束花括号 } **一定**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。

### 6.1. if 、 elseif 和 else

标准的 if 结构如下代码所示，留意 括号、空格以及花括号的位置， 注意 else 和 elseif 都与前面的结束花括号在同一行。

### 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 内的其它语句都 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。

### 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。

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。

### 6.4. 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。

### 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。

## 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，**必须不能**有空格。

参数和变量列表中，逗号前**必须不能**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、逗号、空格以及花括号的位置。