**[PSR规范](http://blog.csdn.net/zhengwish/article/details/51734083)**

**原文：<http://blog.csdn.net/zhengwish/article/details/51734083>**

**原原文：<https://github.com/PizzaLiu/PHP-FIG>**

**本文档是PHP互操作性框架制定小组（PHP-FIG :PHP Framework Interoperability Group）制定的PHP编码规范（PSR:Proposing a Standards Recommendation）中译版。**

目前包括以下几个规范：

PSR-0（弃用）

PSR-1

PSR-2

PSR-3

PSR-4

**1.PSR-0**

**自动加载规范，此规范已被弃用-本规范已于2014年10月21日被标记为弃用，目前新的替代规范为【PSR-4】**

**本文是为自动加载器实现通用自动加载所需要遵循的编码规范**

* 一个标准的命名空间与类名称的定义必须符合以下结构：\<Vendor Name><Namespace>\*(ClassName)
* 其中Vendor Name 为每个命名空间都必须要有的一个顶级命名空间名
* 需要的话，每个命名空间下可以拥有多个子命名空间
* 当根据完整的命名空间名从文件系统中载入类文件时，每个命名空间之间的分隔符都会被转换成文件夹路径分隔符
* 类命名中的每个\_字符也会被转换成文件夹路径分隔符，而命名空间中的\_字符则是无特殊含义的。
* 当从文件系统中载入标准的命名空间或类时，都将添加.php为目标文件后缀
* 组织名称（Vendor Name）、命名空间(Namespace)以及类的名称(Class Name)可由任意大小写字母组成。

**2.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](http://lib.csdn.net/base/php" \o "PHP知识库" \t "http://blog.csdn.net/zhengwish/article/details/_blank)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）**

* PHP代码文件必须以<?php 或<?= 标签开头
* PHP代码文件必须以不带BOM的UTF-8编码 【推荐阅读：[utf-8与utf-8(无BOM)的区别](http://afericazebra.blog.163.com/blog/static/30050408201211199298711/" \t "http://blog.csdn.net/zhengwish/article/details/_blank)】
* 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（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;
* 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

**3.PSR-2**

**代码风格规范，本篇规范是PSR-1基本代码规范的额继承与扩展。本规范希望通过制定一系列规范化PHP代码规范，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。当多名程序员子啊多个项目中合作时，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。**

* 代码必须遵循PSR-1中的编码规范
* 代码必须使用四个空格符而不是tab键进行缩进。
* 每行的字符数应该软性保持在80个内，理论上不可多于120个，但一定不能由硬性限制
* 每个namespace命名空间声明语句和use声明语句块后面，必须插入一个空白行
* 类的开始花名号（{）必须在函数声明后自成一行，结束花名号(})也必须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能有。
* 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后和结束右括号前，都一定不能有空格符。

**4.PSR-3**

**日志接口规范。本规范的主要目的，是为了让日志类库以简单通用的方式，通过接收一个Psr\Log\LoggerInterface对象，来记录日志信息。框架以及CMS内容管理系统如有需要，可以对此接口进行扩展，但须遵循本规范，才能保证使用第三方的类库文件时，日志接口仍能正常对接。**

* LoggerInterface 接口对外定义了八个方法，分别用来记录RFC 5424中定义的八个登记日志：debug、info、notice、warning、error、critical、alert以及emergency。
* 第九个方法-log，其第一个参数为记录的等级，可使用一个预先定义的等级常量作为参数来调用此方法，必须与直接调用以上八个方法具有相同的效果。如果传入的等级常量参数没有预先定义，就必须抛出Psr\Log\InvalidArgumentException类型的异常，在不确定的情况下，使用者不该使用为支持的等级常量来调用此方法。

5.**PSR-4**

**Autoloader 本PSR是关于由文件路径自动载入对应的类的相关规范，本规范是可互操作的。可以作为任一自动载入规范的补充，其中包括PSR-0，此外，本PSR还包括自动载入的类对应的文件存放路径规范。**

* 此处的“类”泛指所有的class类、接口、traits（特点）可复用代码块以及其他类似结构。
* 一个完整的类名需要具有以下结构
* \<命名空间>(\<子命名空间>)\*\<类名>
* 1.完整的类名必须要有一个顶级命名空间，被称为“Vendor namespace”
* 2.完成的类名可以有一个或多个子命名空间
* 3.完整的类名必须有一个最终的类名
* 4.完整的类名中任意一部分中的下划线都是没有特殊意义的
* 5.完整的类名可以由任意大小写字母组成
* 6.所有类名都必须是大小写敏感的
* 当根据完整的类名载入相应的文件......
* 1.完整的类名中，去掉最前面的命名空间分隔符，前面连续的一个或多个命名空间和子命名空间，作为“命名空间前缀”，其必须与至少一个“文件基目录”相对应。
* 2.紧接命名空间前缀后的子命名空间必须与相对应的“文件基目录”相匹配，其中的命名空间分隔符作为目录分割符
* 3.末尾的类名必须与对应的.php为后缀的文件同名
* 4.自动加载器（autoloader）的实现一定不能抛出异常，一定不能触发任一级别的错误信息以及不应该有返回值。