* [0 前言](http://pms.ipo.com/pages/viewpage.action?pageId=9390967" \l "id-技术规范PHP基础代码规范-0前言)
* [1  命名规范](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1命名规范)
  + [1.1  文件名](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.1文件名)
  + [1.2  php标记](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.2php标记)
  + [1.3  类名](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.3类名)
  + [1.4  方法名](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.4方法名)
  + [1.5  变量](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.5变量)
  + [1.6  常量](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.6常量)
  + [1.7  数组定义](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.7数组定义)
  + [1.8  关键字](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.8关键字)
  + [1.9 注](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-1.9注)
* [2  编码风格](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2编码风格)
  + [2.1  文件](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.1文件)
  + [2.2   空行、换行和空格](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.2空行、换行和空格)
    - [2.2.1  操作符](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.2.1操作符)
    - [2.2.2   类](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.2.2类)
    - [2.2.3  函数和方法](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.2.3函数和方法)
    - [2.2.4  简单的语句](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.2.4简单的语句)
    - [2.2.4  控制语句](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.2.4控制语句)
  + [2.3   缩进](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-2.3缩进)
* [3  注释](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-3注释)
  + [3.1  文件注释](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-3.1文件注释)
  + [3.2  类注释](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-3.2类注释)
  + [3.3 方法注释](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-3.3方法注释)
  + [3.4  变量注释](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-3.4变量注释)
  + [3.5  代码块注释](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-3.5代码块注释)
  + [3.6  引入文件注释](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-3.6引入文件注释)
* [4 扩展规范](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-4扩展规范)
* [5 规则之外](http://pms.ipo.com/pages/viewpage.action?pageId=9390967#id-技术规范PHP基础代码规范-5规则之外)

## 1 命名规范

1.1 文件名

文件名参照框架中相关规定。

1.2 php标记

PHP代码使用长标记<?php ?>，禁止使用短标记<?= ?>或其他类型标记。

1.3 类名

类名必须遵循驼峰式命名法，以大写字母开始，如：

class ClassName

{

}

1.4 方法名

驼峰式命名法，以小写字母开始如：

function testFunc()

{

}

1.5 变量

局部变量、类属性、函数和方法的参数，使用驼峰式命名，如：

1、局部变量，函数、方法的参数，使用骆驼命名法命名，以变量类型开头再接变量名的英文含义。如：

function testFunc($sName)

{

}

function testFunc($aUser)

{

}

function testFunc(array $aUser, Car $oCar, $sName = '', $aHouse = array())

{

}

2、全局变量，字母都使用大写，使用"\_"作为每个词的分界，应用中无极端特殊需求不使用全局变量

如：global $USER\_POST\_VARS;

3、变量类型，具体对应关系如下：

变量类型

首字母名称

例如

整型

i

$iNumber = 1;

浮点

f

$fNumber = 2.1;

字符

s

$sDesc = '字符类型';

数组

a

$aArray = array();

对象

o

$oUser = new ClassUser();

布尔

b

$bHave = true;

资源

r

$rResources = fopen("test.txt", "rw");

混合类型

m

临时变量

t

1.6 常量

常量必须使用下划线分割的大写字母，且必须有注释。例如：

const DATE\_APPROVED = 1; //注释

局部常量及有归属的常量应当尽量定义为类常量，特别是具有业务属性的配置值，避免出现＂魔鬼数字＂

1.7 数组定义

$aArray = array(

'a' => 1,

'b' => 2,

'c' => array(

'd' => 1,

'e' => 2

)

);

1.8 关键字

PHP关键字如true，false和null必须用小写。

1.9 注

命名必须具有描述性，但务求简练。不要在变量名中使用长句。通常在变量名中使用一对词语对变量进行简要描述这种方式更好。

类的方法,如外部无调用应当声明为"private"和"protected"的方法，增加\_前缀，反之声明为"public"的方法。

for循环中，自增变量可以使用i、j、k、x、y、z命名，（如非特殊情况，php中请尽量使用foreach语法）

## 2 编码风格

2.1 文件

所有PHP文件必须以<?php标签开头

所有PHP文件必须以一个空白行结束

纯PHP文件,省略关闭标记?>

PHP代码块必须以完整的标记界定

<?php

phpinfo();

模板文件中纯变量输出禁止使用短标记

<?php echo $aData['iAutoID']; ?>

2.2 空行、换行和空格

2.2.1 操作符

在比较运算符（>、<、>=、<=、==、===、!=、<>、!==）、赋值运算符（=）、数学运算符（+、-、\*、/、%）、位运算符（&、|、^、~、>>、<<）、逻辑运算符（!、&&、||）、冒号（:）、问号（?）、字符串连接运算符（.）、字符串连接赋值运算符(.=）前后，以及逗号（,）后请使用空格进行间隔。

例如:

$i = 0;

if (($i < 7) && ($j > 8)){

do\_stuff($i, "foo", $b);

}

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

$i = ($j < $iSize) ? 0 : 1;

}

运算符优先级。对于容易引起迷惑的表达式中不同运算符的优先级，请使用括号来区分优先级。

例如：

$bool = (($i < 7) && (($j < 8) || (4 == $k)))

2.2.2 类

类的左大括号必须从第二行开始，右大括号必须在代码后换行结束，如：

class ClassName

{

}

extends和implements关键字的声明必须和类的名称放在同一行，implements列表可以拆成多行，每一行多要缩进，如：

class ClassName extends ParentClassName implements InterfaceNameA

{

.......

}

2.2.3 函数和方法

方法名和左括号之间必须换行。

左括号和参数之间不能有空格。

多个参数时，逗号之前不能有空格。

多个参数时，逗号之后必须有空格。

例如：

$this->funcName($aRg1, $aRg2, $aRg3);

public function funcName(array $aRg1, array $aRg2, array $aRg3)

{

}

2.2.4 简单的语句

赋值语句，等号前后需加空格，如：$sStr = ‘string’;

自增和自减需要紧接变量名，不能有空格，如：++$iNum;

2.2.4 控制语句

控制结构的关键字后必须有一个空格，控制结构开始的圆括号之后严禁有空格，结束的圆括号之前严禁有空格。

控制结构开始的大括号必须在同一行，结束的大括号必须在代码后换行。

例如：

if (2 == $iNum) {

……

}

if语句

所有条件语句均需使用花括弧，不可省略（即使只有1句语句），条件判断中禁止使用赋值，常量在前变量在后。如：

if (2 == $iNum) {

//code

} elseif (3 == $iNum) {

//code

} else {

//code

}

switch语句

switch语句圆括号前后都没有空格，起始花括号和switch语句在同一行。

case必须缩进，case里的代码必须再有一个缩进。

必须包含default。

如果一个有主体代码的case结构故意的继续向下执行则必须要有一个类似于// no break的注释。

如：

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';

break;

default:

echo 'Default case';

break;

}

for语句

for语句中的条件语句分号后面需带空格，循环体必须使用花括号包含。如：

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

//code

}

2.3 缩进

代码使用4个空格作为缩进，而非用tab作为缩进。

## 3 注释

3.1 文件注释

文件注释置于文件开头用于描述文件的作用与版本信息，如：

/\*\*

\* simple description

\* @desc more description

\* @package \User

\* @author name

\* @date 2015-04-14

\*/

3.2 类注释

类注释置于类开头，主要描述类的作用， 必要时要加实例说明。类的属性应有描述和类型的注释，类本身需要有类名，作者，所属函数库的注释，如@abstract 说明当前类是一个抽象类：

/\*\*

\* simple description

\* @desc description here

\* @author name

\*/

3.3 方法注释

函数及方法注释置于函数或方法开头，主要描述函数或方法的作用，及参数说明，必要时要加使用说明

/\*\*

\* Enter description here...

\* @param int iUid required 用户id

\* @param string sName option 用户姓名

\* @return int

\* /

3.4 变量注释

描述变量信息，与变量申明置于同一行，注释尽可能的对齐，如：

public $sUsername; //用户名

public $sTelphone; //用户电话

private $\_iAge;

3.5 代码块注释

注释的原则是将问题解释清楚，并不是越多越好。

若干语句作为一个逻辑代码块，这个块的注释可以使用/\* \*/方式。

具体到某一个语句的注释，可以使用行尾注释：//

/\*\*

\* 生成配置文件、数据文件。

\*/

$this->setConfig();

$this->createConfigFile(); //创建配置文件

$this->clearCache(); // 清除缓存文件

$this->createDataFiles(); // 生成数据文件

$this->prepareProxys();

$this->restart();

//未完成内容注释，可能后续会完成这个功能。

// @todo author date

// function updUserName()

3.6 引入文件注释

描述引入文件的功能与含义

include(SERVER\_ROOT.’/config/compontentConfig.php’); //模块配置信息， 包含必须的模块配置信息

## 4 扩展规范

框架相关：Laravel代码规范

## 5 规则之外

格式相关

文件编码格式：utf-8；

Tab用4个空格代替；

HTML、CSS名称属性小写；

模板中form表单字段名及URL中的参数名均不得同数据库中字段相同,均使用小写,将字段名去掉数据类型后,从驼峰式修改为下划线分割式；

不得使用@错误抑制符