

1 命名

1.1 代码中的命名不允许中英文混合命名，纯拼音的命名也要尽量避免。

1.2 方法名、参数名、成员变量、局部变量、接口名都统一使用lowerCamelCase风格，必须遵从驼峰形式(首字母小写)。

1.3 控制器，模型，中间件，类名使用驼峰式命名，UpperCamelCase风格(一些通用的大写风格可以例外，但要易于阅读，比如PHPExcel类)。

1.4 数组中的key如果要对数据库的字段名，统一使用全小写加下划线风格，不允许不同风格的key混用。

1.5 常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

1.6 抽象类命名使用Abstract开头，异常类命名使用Exception结尾，测试类命名以它要测试的类的名称开始，以Test结尾，工厂类使用Factory结尾，Interface声明使用Interface结尾。

1.7 模块名统一使用首字母大写，使用单数形式。

1.8 不允许出现任何字符串、数字硬编码，`$status = 0`，应定义常量`define('status', 0)`。

1.9 模块名统一使用首字母大写，使用单数形式。

2 语法

2.1 大括号的使用约定，如果是大括号内为空，则简洁地写成`{}`即可，不需要换行，如果是非空代码块则：

- 缩进统一采用4个空格，不要使用Tab缩进；
- 左大括号前不换行；
- 左大括号后换行；
- 右大括号前换行；
- 大括号后还有else等代码需要换行，表示终止右大括号后必须换行。

例子：

```
if(){  
    /**  
     *TODO
```

```

        *Code Snippet 1
        */
    }
    else{
        /**
        *TODO
        *Code Snippet 2
        */
    }
}

```

2.2 if/for/while/switch/do等保留字与左右括号之间都不要加空格，左括号和后一个字符之间不出现空格，右括号和前一个字符之间也不出现空格,操作符左右两边均留一个空格。

例子：

```

while($name == 'zhangsan'){
    /**
    *TODO
    *Code Snippet 1
    */
}

```

2.3 字符串尽量使用"，以提高效率，需要复杂格式时，再使用""。

2.4 对外暴露的接口签名，原则上不允许修改方法签名，避免对接口调用方产生影响。

2.5 对于覆盖了父类的方法，函数头注释中增加@Override或适当说明。

2.6 对返回值进行比较以及类型转换，必要时使用=== 和 !== 比较""，0，false这样的值。

2.7 不要在你的提交中包含调试代码，就算是注释掉了也不行。

2.8 严格遵守单文件单类的原则。

2.9 适当的时候，提供函数参数的缺省值，这有助于防止因错误的函数调用引起的PHP错误，另外提供常见的备选值可以节省几行代码。

3.0 数组统一采用php5.4以上版本简写形式"[]"（尽量不要用php5.4以下的版本开发）

3.1 尽量使用foreach而不是for

3.2 当判断条件有多个时，使用switch效率比if else更高

3.3 必要时，不要在循环语句中使用查询语句

3 编码

3.1 IDE的text file encoding设置为UTF-8; IDE中文件的换行符使用Unix格式换行符（LF），不要使用windows格式(CR+LF)。

3.2 不要使用字节序标记（BOM），和 UTF-16和UTF-32 不一样，UTF-8 编码格式的文件不需要指定字节序，而且 BOM会在PHP的输出中产生副作用，它会阻止应用程序设置它的头信息。

3.3 起始标签的前面和结束标签的后面都不要留空格，输出是被缓存的，所以如果你的文件中有空格的话，这些空格会在框架输出它的内容之前被输出，从而会导致错误，而且也会导致框架无法发送正确的头信息。

4 注释

4.1 通常情况下，应该多写点注释，这不仅可以向那些缺乏经验的程序员描述代码的流程和意图，而且当你几个月后再回过头来看自己的代码时仍能帮你很好的理解。

4.2 DocBlock风格的注释，写在类、方法和属性定义的前面，可以被IDE识别。

例子：

```
/**
 * Super Class
 *
 * @package Package Name
 * @subpackage Subpackage
 * @category Category
 * @author Author Name
 * @link http://example.com
 */

class Super_class {

    /**
     * Encodes string for use in XML
     *
     * @param string $str
     * @return string
     */
    function encode($str){

    }

}
```

4.3 单行注释应该和代码合在一起，大块的注释和代码之间应该留一个空行，大的代码块之间也应该留空行，形成代码段落。

5 其他

5.1 未完待续，欢迎补充。

Author:KSLab

Date:2018-08-27

