

上海众森计算机科技有限公司



PHP 编程规范

--

提交至:
所有者:
作者: 上海众森计算机科技有限公司 www.zerustech.com
编辑者: Susan Liang, Michael Lee
版本: 1.2
日期: 2007/07/31

免责声明

本文件只用于信息沟通目的，不能作为上海众森计算机科技有限公司对于商品，服务的质量承诺书。上海众森计算机科技有限公司对本文件的一切内容拥有最终解释权。本文件内容如有任何改动，将不做特别通知。

文件修改履历

版本	修改日	修改人	注释
1.0	2007/04/09	Susan Liang	初稿
1.1	2007/05/13	Michael Lee	Modify copyright header information
1.2	2007/07/31	Michael Lee	Modify copyright statement

目录

1 文件.....	4
2 头部.....	4
3 缩进.....	4
4 括号的位置.....	4
5 圆括号，大括号和逗号.....	5
6 字符串.....	5
7 布尔类型和空类型.....	6
8 If 和 while 语句.....	6
9 流程控制.....	6
10 函数.....	9
11 数组.....	9
12 命名.....	9
13 HTTP Post 变量.....	10
14 全局变量.....	11
15 函数.....	11
16 类.....	11
17 注释.....	11
18 附录.....	14
18.1 词汇表.....	14
18.2 引用.....	15

这个文档定义了 eZ Publish 中 PHP 代码的格式和命名规则。为了确保代码库外观一致，eZ Publish 中的代码必须遵循本文的标准。

为了更容易的遵守此标准，我们推荐使用 **emacs** 指令编辑配置文件可以适当的处理缩进和其他外观特征。

1 文件

使用小写字母作为文件名，PHP 文件格式为.PHP。

2 头部

所有的 PHP 文件必须以这个头部开始：

```
// Created on: <nn-Mmm-YYYY HH:MM:NN username>
//
// BUILD VERSION: nnnnn
// COPYRIGHT NOTICE: COPYRIGHT (C) 2003-<CURRENT YEAR> ZERUSTECH LTD. ALL RIGHTS RESERVED.
// COMPANY: ZERUSTECH LTD.
```

3 缩进

缩进为 4 个字符，不要使用 **tab** 键，用空格键来缩进。

4 括号的位置

同一括号的左括号和右括号应该在同一列。

例如：

```
multiply( $x, $y )
{
    // function body
}
if ( x == y )
```

```
{  
    ..  
}  
else if ( x > y )  
{  
    ..  
}  
else  
{  
    ..  
}  
class foo  
{  
    function foo( )  
    {  
        ..  
    }  
}
```

5 圆括号，大括号和逗号

在左括号的后面和右括号的前面要有一个空格。在每个逗号的后面要有一个空格。在中括号内不要有空格，但是在每个类型之间要有一个空格。

空行用来给变量名分组，代码行之间要有联系。同组变量名的缩进量必须是相同的。

空行不仅是显示为空，内容也必须为空，必须删除所有的 **tab** 和空格字符。

如果你不能确定要执行表达式的哪一句，尽量使用更多的圆括号。

6 字符串

PHP 有很多不同的方法来处理字符串。有人倾向于运行速度快的方法，有人倾向于代码的可读性。

PHP 有两种引号 " 和 '。它们都是用来定义一个字符串的。它们的不同之处在于单引号字符串中出现的变量和转义序列不会被变量的值替代。例如：如果你输入 "\$myVariable\n" 和 ' \$myVariable\n'，前者将会输出变量名和换行符，后者将会输出：\$myVariable\n，不做其他处理。因此如果你想提高速度要尽量使用单引号。如果你需要结合文本和变量，可以使用 implode() 函数。这个函数可以由 . 操作符来代替，这样只需分配一次内存，因此更有效率。

下面的例子就是说明怎样在 eZ publish 中使用单引号的。

```
$myHash['key'];  
  
$str = 'This is a string';  
  
$combinedString = implode( ' ', array( 'The color of the ball is ', $color, ' and it costs: ',  
$price ) );  
  
if ( $variable == 'string' )  
{  
    ..  
}
```

7 布尔类型和空类型

PHP 支持任何小写的布尔类型和空类型。

例如：

```
$isSet = true;  
$canPrint = false;  
$unset = null;
```

8 If 和 while 语句

if 语句必须使用 {} 括起来。如果你使用了很多的 if-else 语句，应该用 case 语句来代替，或者使用面向对象的方法重新设计你的代码。if 语句和 while 语句是逻辑语句，在执行时不能分配或更改数据。

9 流程控制

下面的例子是不同 PHP 流程控制的语法。要在每个流程控制名的后面加一个空格。这是和函数不同的地方，在函数名的后面没有空格。PHP 对于流程控制有不同的语法规则，我们更倾向于以下的语法。

```
// if/else statement
if ( $a == $b )
{
    ...
}
else
{
    ...
}

// nested if/else statement
if ( $a == $b )
{
    ...
}
else if ( $a == $c )
{
    ...
}

// while loop
$i = 0;
while ( $i < 100 )
{
    ...

    $i++;
}

// do..while loop
$i = 0;
do
{
    ...
```

```
    $i++;
}
while ( $i < 100 );
// for loop
for ( $i = 0; $i < 100; $i++ )
{
    ...
}
// foreach loop
foreach ( $array as $value )
{
    ...
}
// switch
switch ( $value )
{
    case 0:
    {
        ...
    }break;
    case 'text':
    {
        ...
    }break;
    default:
    {
        ...
    }break;
}
```


10 函数

函数应该尽可能的放在类里面。下面是函数的语法。

```
/*!  
  
    This is my function.  
  
*/  
  
function &myFunction( $paramA = 1, $paramB = 2 )  
{  
  
    return 42;  
  
}
```

11 数组

数组的键值必须使用单引号并且不能有空格。

```
$value = $myHashArray['IndexValue'];
```

12 命名

所有的名称必须为英文且符合语法。应该避免像"foo" 和 "tmp"这类的名称，因为它们没有任何意义。名字中不能包含数字，可以用字母来描述数字。

例如：

```
$ValueOne
```

函数名可以由一些单词构成，所有的单词为小写，但是每个单词的首字母为大写。除此之外，第一个单词的首字母要小写。如果首字母是缩写的一部分，那么整个缩写都要用小写字母。函数返回的布尔值可以命名为一个问题(**has, is, can ...**)。

例如：

```
setText(..);  
  
id();  
  
setID();  
  
getTheValuesForThis();  
  
hasValue();
```

```
isValid();  
canProcess();
```

常量只能由大写字母组成，可以使用_来分隔单词。

例如：

```
define( 'MY_CONSTANT', 42 );
```

成员变量名可以由几个单词组成，每个单词除了首字母大写，其余都小写。不能使用过多单词，正常的是1—3个。只要易理解可以更简洁。

例如：

```
$Len;  
$X, $Y;  
$FirstName;
```

局部变量也可以由几个小写的单词组成，除了第一个单词，每个单词的首字母为大写字母。单词的数量要尽量少，可以缩写单词。

```
$index, $x, $y, i;  
$xAdd;  
$firstName, $lastName;
```

参数的命名同局部变量

类的名称可以由几个单词组成，每个单词都要小写，除了首字母要大写。类的名称必须以同一个字符串开始，例如eZ，用来唯一标识类。

例如：

```
eZUser  
eZContentObject  
eZImageVariation
```

要避免使用全局变量，把它们放到一个类中或设置为静态变量。

13 HTTP Post 变量

POST 变量要遵循以下语法：

```
Class_FirstName  
Class_ValueOne
```

```
Content_CategoryID  
Content_LongVariable  
SearchValue
```

类和内容名要基于变量并且为指定的某一模块，函数或视图的变量。像 **SearchValue** 这样的变量应该用于全局，尽管你也可以使用 **Global** 作为基础。

14 全局变量

通过使用类名作为名称的基础，全局变量名应该是唯一的。例如类的变量和全局列表。

```
$instance =& $GLOBALS['eZUserInstance'];  
$handlers =& $GLOBALS['eZCodecHandlers'];
```

15 函数

除非有好的理由，所有的函数应该放到类中。

函数应该简短并只做一件事。函数的长度取决于函数的复杂度，越复杂的函数应该越短。如果过于复杂则把函数分开成辅助函数。

函数中局部变量的数量不能大于 7。7 是人们可以同时控制事物数目的极限。如果函数过大，将它分开。

16 类

类有一个规定。所有的成员变量都是私有的。辅助函数也应该是私有的。不访问成员变量的函数应该为静态的。

17 注释

使用注释应该遵守 **Doxygen** 工具的语法，这个工具用来生成类的参考文档。

```
/*! \defgroup eZGroup Grouped classes  
  
    Use groups to place common classes together.  
*/  
/*!  
/*!
```

```
\class eZMyClass ezmyclass.php

\brief eZMyClass handles ...

\ingroup eZGroup

This is a more complete description of the class.

Here is an code example:

\code

$myObj = new MyClass();

\endcode

See also these classes:

\sa eZUser, eZContentClass

\todo Add this function..

\bug Does not handle...

*/

class MyClass
{
    /*!

        This is the contructor of MyClass. It handles....

    */

    function MyClass()
    {
        ...
    }

    /*!

        \private

        This is a private helper function.

    */

    function doSomething()
```

```
{
    ...
}

/*!
 \return the value of ...
 */

function value()
{
}

/*!
 \static

This can be called like MyClass::doIt()
 */

function doIt()
{
    ...
}

///// \privatesection
/// This is a variable description.
var $MyVar;
}
```

行内注释

何时使用行内注释？这有一些指南，指导你该怎么做。

行内注释大概的意思就是解释这一行代码是做什么的，但是会有一些问题。

- 隐藏信息。它们经常对使用系统 API 的开发者隐藏。例如被认为是有效输入非公有的关键信息。
- 当注释和代码混在一起时会变得很难阅读。
- 在很多情况下，注释是不好或者难读密码的藉口。

在写行内注释时应该考虑的问题。

- 是否能为这个类或函数起一个更好的名字来反映它是做什么的？
- 是否可以更改变量名或执行顺序来更好反映正在做什么？
- 行内注释时候包含应该在 API(doxxygen)文档中解释的重要信息？
- 代码是否能被简化或分离以便于更容易阅读和理解？

大多数情况下，考虑以上问题将帮助你避免使用行内注释。但是，在某些情况下，它们是必须的。典型的元素必须要有行内注释的。

- 常规表达，了解 RegExp 花费的时间。这时注释可以帮助理解。
- Internal hacks，处理性能或 PHP 的特殊问题。

单独一行的行内注释是不够的，有时最好在 API 文档中包含一些相同的文本。例如在一个特定版本的 PHP 中使用工作区。

行内注释书写时只能写到行尾，像这样。

```
// A workaround for PHP 4.2.2 where the xyz functionality is broken
if ( php_version() == '4.2.2' )
{
    ...
}
```

18 附录

18.1 词汇表

词汇	解释

附录 1: 词汇表

18.2 引用

引用	注释
eZ Publish 官方站点	http://ez.no/ezpublish/documentation/development/standards/php
上海众森计算机科技有限公司	www.zerustech.com

附录 2: 引用