



## Xoops 模块开发编码标准

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops模块开发_编码标准)
- ✓ 本电子书整理自站长百科[Xoops模块开发\\_编码标准](#)词条，查看内容请访问网站。

**好消息：** Lunarpages再推新优惠码：*1un50off* 所有方案减50美元

### 概述

这份文档为开发者, 工作组以及XOOPS项目提供了编码标准和指导. 主要涉及到以下内容:

- [PHP](#) 文件格式
- 命名规范
- 编码风格规范
- 代码内文档化
- 错误与异常

开发人员的角色:

- X00PS 开发人员: 这些开发人员来维护X00PS核心[程序](#), 其中包括:

核心开发人员: 对 X00PS 核心代码的维护, 这些核心代码将在 X00PS 应用程序开发人员那里得到更好的应用.

框架开发人员: 贡献出自己的开发框架, 有机会集成到 X00PS 内核当中.

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科[Xoops\\_模块开发\\_编码标准](#)词条，查看内容请访问网站。

前端开发人员：进行 XOOPS 的风格主题和模块的内容模板的设计。

- XOOPS 应用程序开发人员：开发者在 XOOPS 的基础上开发自己的应用程序

模块开发人员：开发者基于 XOOPS 平台以及类库开发第三方应用模块。

## 目标

编码规范对于任何一个开发项目都是很重要的,尤其是多人协作的情况,良好的编码规范可以为我们带来高质量的代码,较少bug,以及便于维护。

我们力争做到:

- 尽可能的简单
- 更为方便,例如方法的说明,常量的定义,以及格式这些都可以通过开发工具的自动补全功能来实现。

当考虑上述的目标,每一种情况需要考虑实际情况进行取舍。

## PHP 文件格式要求

### 常规

对于只包含有 PHP 代码而无别的代码的文件,可以不使用结束标记“?>”,要注意的是,在 XOOPS 的开发中代码结束必须要带有结束标记“?>”。

### 制表符

缩进的时候使用 4 个空格(SPACE)字符(键),不要使用 TAB 字符(键)。在编辑器中可以设置 TAB 字符以 4 个空格代替,目的是为了不同操作系统下的表现一致。

### 每行最大字符数

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops 模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

每行不要超过 80 个字符，即，在实际开发过程中每行的代码尽量不要超过第 80 列这个边界，稍微超过也可以，PHP 代码每行最大字符数是 120。

## 行结束符

行结束符采用 [UNIX](#) 文本文件标准，即每行以“换行符 (LF)”结束，目的是为了不同操作系统下的表现一致。在 [ASCII](#) 码中，换行符是的十进制值为 10，十六进制值为 0x0A。

不要使用 [Macintosh](#) 系统的标准：以回车符 (CR，十六进制值是 0x0D) 为行结束符。

也不要使用 [Windows](#) 系统的标准：以回车/换行 (CRLF, 0x0D, 0x0A) 为行结束符。

每行的结尾不要包含多余的空格，为了保证这个规范，很多编辑器都可以进行一些配置，在保存的时候自动去除多余的空格。

## 命名规范

### 总体要求

1. XOOPS 核心中所有的类和函数命名必须以 Xoops 作为前缀；

- 函数名称首字母要小写；下列情况要用 “\_” 隔开而不是驼峰式写法：

核心函数必须以 xoops\_ 为前缀

框架以及其它类库 必须以 xoops\_[框架或类库标记符]\_ 为前缀

- 类名字要用驼峰式写法；

2. XOOPS 核心中所有的变量定义必须以 \$xoops 作为前缀，并且采用驼峰式写法。

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条，查看内容请访问网站。

- 私有变量或者局部变量也要采用驼峰式写法
3. 第三方应用程序, 包括模块, 不要以 Xoops\_ 作为前缀, 但是要有统一的标识符前缀
- 块中类的命名必须以 [模块标记符, 通常为模块目录名称] 作为前缀如:  
NewbbPost
  - 模块中函数命名必须以 [模块标记符, 通常为模块目录名称]\_ 作为前缀, 如:  
newbb\_getPostCount ()
  - 模块中变量命名必须以 \$[模块标记符] 作为前缀, 如: \$newbbPostCount

## 应用程序接口 (API)

当程序开发人员创建一个应用程序接口时, 如果使用合成词来定义, 使用下划线来隔开, 而不是使用驼峰式写法. 如果需要使用一个字符串, 通常是采用小写字面, 最好定义一个常量去使用.

## 类

系统, 内核以及框架中类的命名是带有 Xoops 的, 如 XoopsUser, XoopsCaptcha. 模块中类命名是以[模块标识符]开头, 如 NewbbPost.

## 接口

接口类的命名规则与其他类的命名规则类似, 必须以单词“Interface”结尾如:

XoopsLogger\_Interface

## 文件命名

对于所有文件, 只有字母、数字、下划线和短划线 (“-”) 可以使用, 不允许使用空格和其他特殊字符.

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

只要是包含有 PHP 代码的文件都必须以“.php”为扩展名.

文件名称必须为小写.

## 目录命名

对于所有目录, 只有字母、数字、下划线和短划线 (“-”) 可以使用, 不允许使用空格和其他特殊字符.

目录名称必须为小写.

## 函数与方法的命名

内核, 系统模块中函数的命名遵循 `xoops_doSomething([...])` 格式.

框架中函数的命名以 `xoops_[小写标识符]_` 作为前缀, 如 `xoops_pear_doSomething([...])`.

模块中函数的命名以 `[小写模块标识符]_` 作为前缀, 如 `newbb_getTopic([...])`.

- 函数名只能包含字母, 数字字符和下划线, 虽然允许使用数字, 但并不提倡.
- 函数名必须以小写字母开始.
- 为增强代码的可读性, 允许命名一定程度的详细和冗长.
- 对于面向对象编程, 对象的存取器总是以“get”或“set”为前缀.
- Class methods that are declared as protected or private are encouraged to follow same style as public methods although it is desired by some external frameworks to begin with a single underscore, i.e. without leading single underscore.

✓ 出处: 站长百科

✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)

✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

- 本规范不允许有全局范围的函数（即游离于对象之外的函数），这些函数应该包装进一个静态类.
- 在一个类中声明静态方法和变量时要用 “static” 不要用 “private” 反之亦然，如果使用 “final” 则这个方法将不能扩展.

## 参数

使用 “NULL” 代替默认值 “FALSE”，如：

```
public function foo($required, $optional = NULL)
```

当 \$optional 没有或不需要指定一个特定的默认值.

但是，如果可选参数是一个布尔值，其默认的逻辑为 “TRUE” 或者 “FALSE”，可以使用 “TRUE” 或者 “FALSE”.

## 变量

- 系统全局变量必须以 \$xoops 为前缀, 如 \$xoopsConfig
- 模块全局变量必须以 \$[小写模块标记符] 为前缀，并且遵循驼峰式写法，如 \$newbbPostCounter
- 第三方框架类库中的变量定义
- 变量名只能包含字母和数字字符，也允许使用下划线和数字但不提倡的.
- 在类中声明为 public 的成员变量则不允许使用下划线.
- 对于声明为 private 或 protected 的类成员变量，命名规范与成员变量一样，即变量名不必使用一个下划线开头.
- 函数命名, 变量命名必须以小写字母开始，而之后的单词，遵循首字母大写的驼峰写法（camelCaps）.

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops 模块开发\\_编码标准](#) 词条，查看内容请访问网站。

- 为了增强可读性,变量命名同样需要一定程度的详细和冗长,并且要有实际意义。类似于*\$i*、*\$n* 这样的变量命名,除了用于较少代码的循环语句,在其他情况下是不允

许的。如果循环语句超过 20 行代码,那么用于循环的索引变量也应该使用有意义的名字。

## 常量

### Variables

系统常量以 XOOPS\_ 做为前缀,如 XOOPS\_URL

框架常量以 XOOPS\_[标记符]\_ 做为前缀 如 XOOPS\_PEAR\_CONSTANT

模块中常量以 [模块标记符]\_ 做为前缀如 NEWBB\_CONSTANT

#### ' 语言常量定义规范

必须以下划线开头,如: `_XOOPS_LANGUAGE_CONSTANT`,  
`_NEWBB_LANGUAGE_CONSTANT`

- 常量名允许使用字母和数字字符,与函数命名和变量命名等不同的是,他也允许使用下划线,对数字的使用也没有限制.
- 常量的所有字母都必须大写.
- 为了增强可读性,单词之前以下划线分开. 如,

`"XOOPS_EMBED_SUPPRESS_EMBED_EXCEPTION"` 不要写成  
`"XOOPS_EMBED_SUPPRESSEMBEDEXCEPTION"`.

- 常量必须使用 `const` 指示符定义为类成员,在全局范围使用 `define` 定义常量虽然允许,但也不提倡.

✓ 出处: 站长百科

✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)

✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

## 布尔值与 NULL

如同 php 手册中所述, XOOPS 同样使用大写字母来编写.

## 模块模板名称定义规范

## 编码风格

### PHP 代码标记符

在 XOOPS 中必须使用完整的 PHP 标记符, 标准的 PHP 标记符为:

```
<?php
```

```
?>
```

不允许使用简写标记符.

### 字符串

#### 纯文字字符串

可以使用双引号和单引号来标记字符串. 对于纯文字字符串 (不包含变量替代), 必须使用单引号包含:

```
$a = 'Example String';
```

#### 字符串中的特殊字符

当字符串中包含撇号 (') 和单引号时, 可以使用双引号. 这种方式多用于 SQL 语句的书写:

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops 模块开发\\_编码标准](#) 词条, 查看内容请访问网站。



```
$sql = "SELECT `id`, `name` from `people` WHERE `name`='Fred' OR  
`name`='Susan'";
```

这样的语法更加清晰.

含有变量替代的字符串

变量的替换推荐用大括号标记.

推荐:

```
$greeting = "Hello {$name}, welcome back!";
```

不推荐:

```
$greeting = "Hello $name, welcome back!";
```

为了保持一致性, 下列方式也是不推荐的:

```
$greeting = "Hello ${name}, welcome back!";
```

字符串连接

多个字符串的可以使用 "." 符号连接. 使用 "." 符号时必须在其前后加入一个空格, 这样可以增强可读性如:

```
$project = 'Xoops' . ' ' . 'Project';
```

当使用 "." 操作符连接多个字符串时, 可以把语句拆分成多行来增强可读性, 对于这种情况, 每个后续行应该使用空格填充, 使 "." 操作符刚好对齐在 "=" 下面:

```
$sql = "SELECT `id`, `name` FROM `people` "
```

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/Xoops> 模块开发 编码标准
- ✓ 本电子书整理自站长百科 [Xoops 模块开发 编码标准](#) 词条, 查看内容请访问网站。

```
. "WHERE `name` = 'Susan' "  
  
. "ORDER BY `name` ASC ";
```

## 数组

### 数字索引数组

数组的索引不允许使用负数。

索引数组可以使用任何非负整数开始，但并不提倡这么做，建议每个数组都应该使用索引 0 开始。

当使用 array 关键字定义索引数组时，每个逗号后面都应该跟随一个空格字符，目的是增强可读性：

```
$sampleArray = array(1, 2, 3, 'XOOPS', 'Project');
```

使用 array 关键字定义索引数组的语句可以拆成多行，这种情况下每个后续行都应该使用空格填充头部，使每行以如下的形式保持对齐：

```
$sampleArray = array(1, 2, 3, 'XOOPS', 'Project',  
                    $a, $b, $c,  
                    56.44, $d, 500);
```

### 关联数组

当使用 array 关键字定义关联数组时，建议把语句拆分成多行，对于这种情况，应该使用空格使每一行的键和值分别对齐：

```
$sampleArray = array('firstKey'      => 'firstValue',
```

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条，查看内容请访问网站。

```
'secondKey' => 'secondValue');
```

## 类

### 类的声明

类的声明应该遵守以下要求:

- 大括号必须写在类名字的下一行.
- 每个类都必须有一个遵守 PHPDocumentor 标准的注释文档块.
- 类内部的代码都必须缩进 4 个空格.
- 一个 PHP 文件只允许有一个类.
- 在一个类文件里可以放置其他代码, 但不提倡, 对于这种情况, 必须使用 2 个空行, 把类代码和其他 PHP 代码分开.

下面是一个规范的类的声明:

```
/**  
  
 * 类说明的注释块  
  
 */  
  
class XoopsClass  
  
{  
  
    // 类的内部代码  
  
    // 必须缩进 4 个空格  
  
}
```

### 类成员变量

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

成员变量的命名必须遵守变量命名规则.

类成员变量的声明必须位于类的顶部, 在函数定义之前.

不允许使用var [关键字](#), 成员变量的声明必须使用关键字: private、protected 或者public。尽管可以通过把变量声明为public, 以便直接访问成员变量, 但本规范推荐使用get/set存取符来访问变量.

## 函数与方法

函数与方法的定义

函数命名必须遵循命名规范.

类内部的函数必须使用 private、protected 和 public 等关键字, 表示该函数的可见性.

声明静态变量的时候, 众多 PHP 社区习惯写在类的最前面:

```
public      static foo()    { ... }
```

```
private     static bar()    { ... }
```

```
protected  static goo()    { ... }
```

函数里大括号的用法与类一致, 即大括号必须位于函数名字的下一行, 函数名字与括号之间没有空格.

下面是规范的类成员函数的书写方法:

```
/**
```

```
 * 类说明文档的注释块
```

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

```
*/  
  
class XoopsFoo  
  
{  
  
    /**  
  
     * 方法说明文档的注释块  
  
     */  
  
    public function sampleMethod($a)  
  
    {  
  
        // 函数的内部内容  
  
        // 必须缩进 4 个空格  
  
    }  
  
    /**  
  
     *方法说明文档的注释块  
  
     */  
  
    protected function _anotherMethod()  
  
    {  
  
        // ...  
  
    }  
  
}
```

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops 模块开发\\_编码标准](#) 词条，查看内容请访问网站。

```
    }  
}
```

返回值 (return) 不允许使用括号. This can hinder readability and can also break code if a function or method is later changed to return by reference.

```
function foo()  
  
{  
  
    // 错误的写法  
  
    return($this->bar);  
  
    // 正确的写法  
  
    return $this->bar;  
  
}
```

The use of type hinting is encouraged where possible with respect to the component design. For example,

```
class XoopsComponent  
  
{  
  
    public function foo(SomeInterface $object)
```

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

```
{  
  
    public function bar(array $options)  
    {  
  
    }  
}
```

## 函数和方法的用法

禁止使用全局函数.

如果函数有多个参数,需要在每个逗号后面添加一个空格,如下例所示:

```
threeArguments(1, 2, 3);
```

调用阶段不允许传递引用传递参数,而应该把他放在函数定义阶段.

对于允许使用数组参数的函数,函数调用允许使用 array 声明语句,并且允许分割成多行,同时需要通过缩进保持可读性,例如下面的例子:

```
threeArguments(array(1, 2, 3), 2, 3);  
  
threeArguments(array(1, 2, 3, 'XOOPS', 'Project',  
                    $a, $b, $c,  
                    56.44, $d, 500), 2, 3);
```

## 控制语句

If / Else / Elseif

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

控制语句中 if 和 elseif 关键字之后，必须一个空格与后面的左括号分割，右括号后面也必须有一个空格。在括号里面的条件语句，操作符两边必须有空格以保持可读性，如果括号里的条件较多，建议根据逻辑分组通过添加括号。左大括号应该写在条件语句的同一行，而右大括号应该独自放在一行，括号内部的内容应该缩进 4 个字符。

```
if ($a != 2) {  
  
    $a = 2;  
  
}
```

对于包含有 elseif 或 else 的 if 语句，其格式要求参照如下例子：

```
if ($a != 2) {  
  
    $a = 2;  
  
} else {  
  
    $a = 7;  
  
}
```

```
if ($a != 2) {  
  
    $a = 2;  
  
} elseif ($a == 3) {  
  
    $a = 4;  
  
} else {
```

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/Xoops> 模块开发 编码标准
- ✓ 本电子书整理自站长百科 [Xoops 模块开发 编码标准](#) 词条，查看内容请访问网站。



```
$a = 7;  
  
}
```

尽管 PHP 允许在某些情况下这些语句里可以不使用大括号, 但我们的编码规范里不允许这么做, 所有的 if、elseif 和 else 语句都必须使用大括号.

## Switch

switch 内部的内容不用缩进, 每个 case 语句下的内容也同样缩进 4 个空格.

```
switch ($numPeople) {  
  
case 1:  
  
    break;  
  
case 2:  
  
    break;  
  
default:  
  
    break;  
  
}
```

switch 语句中都必须有一个 default 语句, 不能省略.

## 内部文档化

- ✓ 出处: 站长百科
- ✓ 原文地址: [http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops 模块开发\\_编码标准](#) 词条, 查看内容请访问网站。

## 文档格式

所有的文档块（即 doc blocks）都必须遵循 php Documentor 格式

所有为 Xoops 平台写的或者应用在 Xoops 平台的源代码文件，都必须在每个文件顶部包含文件级的文档块，以及在每个类定义的上边包含类级的文档块。

不能用 '#' 作为注释的标记符。

## 文件中的文档块

任何包含 PHP 代码的文件都必须在其顶部包含文档块，并至少包含以下 php Documentor 标记：

```
/**
```

```
* 关于本文件的简要说明
```

```
*
```

```
* 关于本文件的详细描述（如果有的话）...
```

```
*
```

```
* 许可信息
```

```
*
```

```
* You may not change or alter any portion of this comment or credits
```

```
* of supporting developers from this source code or any supporting source  
code
```

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条，查看内容请访问网站。

\* which is considered copyrighted (c) material of the original comment or credit authors.

\*

\* @copyright The Xoops Project  
<http://sourceforge.net/projects/xoops/>

\* @license <http://www.fsf.org/copyleft/gpl.html> GNU public license

\* @author 作者名称 <author email, or website>

\* @version \$Id\$

\* @since File available since Release 3.0.0

\*/

## 类中的文档块

每个类级别的文档块都必须至少包含以下 php Documentor 标记:

/\*\*

\* 类的简单说明

\*

\* 类的详细说明 (如果有的话)...

\*

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/Xoops> 模块开发 编码标准
- ✓ 本电子书整理自站长百科 [Xoops 模块开发 编码标准](#) 词条, 查看内容请访问网站。

```
* @copyright   The XOOPS Project
http://sourceforge.net/projects/xoops/

* @license     http://www.fsf.org/copyleft/gpl.html GNU public
license

* @author      Author Name <author email, or website>

* @version     $Id$

* @since       File available since Release 3.0.0

*/
```

## 函数中的文档块

每个函数，包括对象方法，都必须包含至少下列文档块：

- 函数功能描述
- 所有的参数
- 所有可能返回的值

```
/**
```

```
* 关于函数的说明
```

```
*
```

```
* @param Place $where Where something interesting takes place
```

```
* @param integer $repeat How many times something interesting
should happen
```

```
* @return Status
```

- ✓ 出处：站长百科
- ✓ 原文地址：[http://www.zzbaike.com/wiki/Xoops\\_模块开发\\_编码标准](http://www.zzbaike.com/wiki/Xoops_模块开发_编码标准)
- ✓ 本电子书整理自站长百科 [Xoops\\_模块开发\\_编码标准](#) 词条，查看内容请访问网站。

```
*/  
  
public function xoops_doSomethingInteresting(Place $where, $repeat = 1)  
  
{  
  
    // implementation...  
  
}
```

## 更多电子书

SEO 方面的知识有很多，对于新手来说，如果你不知道，不清楚这方面的知识，那么，你可以看看这两部搜索引擎指南：

SEO搜索引擎优化基础教程：<http://bbs.zzbaike.com/thread-9952-1-1.html>

SEO搜索引擎优化高级教程：<http://bbs.zzbaike.com/thread-12692-1-1.html>

wordpress的中文翻译文档：<http://www.wordpress.la/codex.html>，wordpress 开发的相关知识都有，对WordPress开发感兴趣的博主会有一些的用处。

WordPress免费主题：<http://www.wordpress.la/theme.html>

WordPress免费插件：<http://www.wordpress.la/plugin.html>

WordPress主题制作电子书：<http://bbs.zzbaike.com/thread-9954-1-1.html>

Apache2.2 中文文档电子书：<http://bbs.zzbaike.com/thread-9955-1-1.html>

IXWeHosting 控制面板使用手册(在线版+PDF 电子书)：

<http://bbs.zzbaike.com/thread-9953-1-1.html>

更多电子书下载：<http://down.zzbaike.com/ebook/>

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/Xoops> 模块开发 编码标准
- ✓ 本电子书整理自站长百科 [Xoops 模块开发 编码标准](#) 词条，查看内容请访问网站。

1. 美国主机 IXWebHosting 使用视频教程（在线观看及下载）

<http://bbs.zzbaike.com/thread-47008-1-1.html>

2. Godaddy 主机及域名使用视频教程（在线观看及下载）

<http://bbs.zzbaike.com/thread-50005-1-1.html>

如果您有站长类电子书，请到这里与我们分享：

<http://bbs.zzbaike.com/forum-69-1.html>

详情见：<http://bbs.zzbaike.com/thread-23156-1-1.html>

站长百科感谢您下载阅读，多谢支持！

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/Xoops> 模块开发 编码标准
- ✓ 本电子书整理自站长百科 [Xoops 模块开发 编码标准](#) 词条，查看内容请访问网站。