

# Wordpress 主题制作简明教程

Wordpress 教程网提供



[www.wpcourse.com](http://www.wpcourse.com)

时 间 2009-10  
作 者 [tomheng](#)  
版 本 V1.0

# 目录

简单说明

基本流程

制作博客的 **header**

制作博客的 **footer.php**

制作博客的 **index.php** 文件

制作博客的 **sidebar.php** 文件

# 第一讲 简单说明

老伍注：很多人开始使用 *wordpress* 都是因为看到漂亮的 *wordpress* 主题。为了能让每一个 *WordPress fans* 都能制作个性化的主题，我们经过精心的准备，终于推出了这套 *WordPress* 主题制作入门教程。这套教程主要由 *tomheng* 编写，老伍修正，希望大家能从中对 *wordpress* 主题制作有一个基本的了解。

*WordPress* 是一个很优秀的开源博客程序，受到全世界网民的追捧。它不但是一个简洁易用的博客应用程序，同时也是个学习的好工具。通过主题的制作你可以学习网页设计相关知识，学习插件开发则可以提高你 *php* 编程的能力，如果更深入的了解一下 *WordPress* 的框架结构，那对你架构设计能力也是一种提高。我们提供的这套入门教程旨在普及 *WordPress* 主题制作的基础知识，面向最基础的入门用户，让大家认识到制作 *WordPress* 主题是简单的，有趣的（*It is easy, it is fun* 这是我们的口号）。这个简单的教程会讲解如何制作 *WordPress* 主题的基本步骤，但是制作漂亮的主题还需要更多的经验和努力才行。

学习我们的教程需要您首先的是一个 *WordPress* 的熟练的用户，如果你还不是很熟悉如何使用 *WordPress*，请移步 *WordPress* 教程网查看 *WordPress* 的基本的使用教程。

教程开始之前还是要说一些网页设计的基本的规范以及一些基本的 *php* 常识：

## （1）所有 *html* 标签必须关闭

<ul></ul>

## （2）*Html* 标签的不能交叉嵌套

<div><span></div></span> 这样是不允许的

<div><span></span></div> 这样是正确的

## （3）凡是在<?php ...?>或者在<? ...?>里的内容都是 *php* 代码

## （4）*Php* 的两种注释方法

A、// 单行注释          B、/\*\*/ 跨行注释

即以/\*开头到\*/终止的中间部分全部为注释代码

## 5) *css* 样式文件的注释方法 只有一种既 */\*\*/* 跨行注释

推荐一些工具和网站

1) *php* 不懂得问题去。 *Php.net* 查询

2) 与 *css*, *html* 有关的问题去: *w3school* 查询

## 第二讲 主题制作基本流程

没错制作主题教程是简单的也是有趣的。上一讲我们对本 [wordpress 主题教程](#) 进行了简单说明，现在就让我们先来配置一下我们的开发环境吧。

(1) 在自己的电脑安装 **WordPress** 运行的环境（在本机搭建 [WordPress 的安装环境 XAMPP](#)==[WordPress 教程网教程系列](#)）；

安装过程中可能会遇到一些问题，一般就是迅雷等软件占用80端口。所以安装之前最好把迅雷给关了。另外 [phpnow](#) ,[appserve](#) ,[nertrigo](#) 等都可以完成 [xampp](#) 的工作。

(2) 在本机服务器安装 **WordPress**（在本机安装 [WordPress 全过程](#) ==[WordPress 教程网教程系列](#)）。

这个相对简单一些，如果一切顺利这时候，在自己的浏览器输入 <http://localhost> 就可以看到自己装的 **WordPress** 啦。

(3) **推荐：**准备一些软件，[dreamweaver](#) ，[fireworks](#) ,最好装两个以上的浏览器（包含 [ie](#) , [firefox](#)）当然这些不一定用的着，只是推荐而已，大家根据自己的知识水平准备一下就可以啦。

### 现在万事俱备，接下来就开始介绍教程制作的相关知识。

(1) 主题的存放位置。

所有的主题都存在于：你的博客所在目录 `/wp-content/themes` 下，每一个文件夹就是一个主题。所以我们制作主题的第一步就是在这个目录建立一个文件夹，文件夹的名字可以用主题的名字，您还得先为主题取个好听的名字奥。比如我们就建立一个叫 `wpc-tomheng` 的文件夹。

到 [wordpress](#) 后台看一下是不是存在我们的主题。令人失望的是没有看到我们的主题，这是什么问题呢？其实建立一个文件夹只是给我们的主题买了块地皮而已，我们还需要在上面建立一些东西，才能在后台看到的。要在后台看到我们的主题最主要的一步就是在主题目录（我这里是 `wpc-tomheng` ,要根据自己的具体情况在相应的目录里，以后我们就直接以 `wpc-tomheng` 作为说明，您在操作的时候要具体转化为自己的目录。）

建立一个叫 `style.css` 的文件。这个文件是用来存放 `css` 样式文件，这里面也包含了关于主题的一些信息。与主题相关的信息都是放在 `/*主题相关的信息*/` 里面。我们在里面写入最简单的一个信息-关于主题的名称。在 `style.css` 文件里写入如下信息：

```
/*
```

```
Theme Name: wpc-tomheng
```

```
*/
```

**说明：**在这里可以填入更多的相关信息，但是都要符合 [WordPress](#) 的规范才行。这些信息不是必要，但是我们最好填写一些必备的信息（如：主题的名字，作者，等信息），这样看起来更规范。保存文件。到后台看一下，我们依然没有发现我们的主题，我们的主题却被列入了已损坏的主题里面，提示信息为：缺失模版文件。是的我们的主题确实小但是五脏并未俱

全，接下来再建立一个模版文件就好啦。在 `wpc-tomheng` 目录建立一个 `index.php` 的文件，这时候我们的主题能在后台的主题选项目录看到，同时我们的主题也可以安装了。

这就是一个新的主题必须具有的最基本的结构（`style.css` 和 `index.php` 文件）。现在我们安装上自己制作的主题，但是我们到前台查看自己的博客时候，什么内容也没有看到。这是为什么呢？

当然，我们的工作刚刚起步。我们还需要给我们的主题继续的添加东西，才可以在前台有个完美的展示。在 `index.php` 里写入任何的 `html` 标签或者简单的文本都可以在前台显示出来。You may wan to try it .

但是我们怎么把我们在博客里的文章或者其他的信息显示出来呢，这就要靠我们的 WordPress template tags

[http://codex.wordpress.org/Template\\_Tags](http://codex.wordpress.org/Template_Tags)

我们在这里先介绍一个最简单的 `tag`，（在开始的几讲里我们只是介绍一下主题的最基本知识 and 相关的流程，关于更详细的教程我们会在后续的章节发布出来，请密切关注 `wpc` 的动向）。

在 `index.php` 里写入如下代码：

```
<html>

<body>

<?php
bloginfo('name');
?>

</body>

</html>
```

注：请注意其中的引号，有可能您直接 `copy` 过去的话，会变成全角的，应该是半角英文的引号。

现在在博客的首页应该可以看到自己博客的名字了，这就是 `tag` 的一个最简单的应用。

现在 `wpc-tomheng` 应该算是一个完整的主题，但绝不是完美的主题。因为他基本没有展现我们博客的内容，也没有漂亮的外观。展现博客的内容要靠 `template tags`（例如 `bloginfo()`）来完成，改变外观要靠 `style.css` 来完成。这也是制作主题的关键，最能展示个人创意的地方。

接下来，如果您愿意把自己制作的主题 `release` 出来的话，就可以打包成 `.zip` 文件，然后提交到 WordPress。

恩，现在是不是对 WordPress 主题制作有个基本的了解了，接下来我们会详细的讲解 `template tags` 的相关知识。

这个文件是用来存放 `css` 样式文件，这里面也包含了关于主题的一些信息。与主题相关的信息都是放在 `/*主题相关的信息*/` 里面。我们在里面写入最简单的一个信息-关于主题的名称。在 `style.css` 文件里写入如下信息

```
/*  
Theme Name: wpc-tomheng  
*/
```

说明：在这里可以填入更多的相关信息，但是都要符合 WordPress 的规范才行（[http://codex.wordpress.org/Theme\\_Development](http://codex.wordpress.org/Theme_Development)）这些信息不是必要，但是我们最好填写一些必备的信息（如：主题的名字，作者，等信息），这样看起来更规范。保存文件。到后台看一下，我们依然没有发现我们的主题，我们的主题却被列入了已损坏的主题里面，提示信息为：确实模版文件。是的我们的主题确实小但是五脏并未俱全，接下来再建立一个模版文件就好啦。在 wpc-tomheng 目录建立一个 index.php 的文件，这时候我们的主题能在后台的主题选项目录看到，同时我们的主题也可以安装了。

这就是一个新的主题必须具有的最基本的结构（style.css 和 index.php 文件）。现在我们安装上自己制作的主题，但是我们到前台查看自己的博客时候，什么内容也没有看到。这是为什么呢？

当然，我们的工作刚刚起步。我们还需要给我们的主题继续的添加东西，才可以在前台有个完美的展示。在 index.php 里写入任何的 html 标签或者简单的文本都可以在前台显示出来。You may wan to try it.

但是我们怎么把我们在博客里的文章或者其他的信息显示出来呢，这就要靠我们的 WordPress template tags ([http://codex.wordpress.org/Template\\_Tags](http://codex.wordpress.org/Template_Tags))。我们在这里先介绍一个最简单的 tag，（在开始的几讲里我们只是介绍一下主题的最基本知识 and 相关的流程，关于更详细的教程我们会在后续的章节发布出来，请密切关注 wpc 的动向）。

在 index.php 里写入如下代码：

```
<html>  
<body>  
  <?php  
    bloginfo('name');  
  ?>  
</body>  
</html>
```

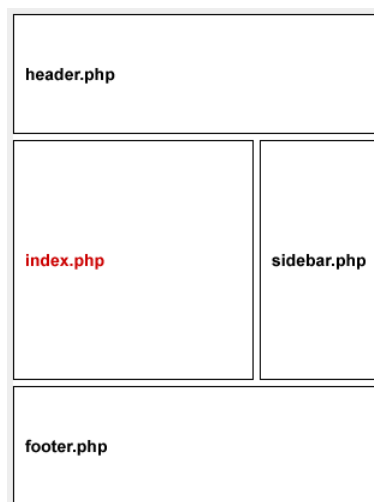
现在在博客的首页应该可以看到自己博客的名字了，这就是 tag 的一个最简单的应用。

现在 wp-tomehng 应该算是一个完整的主题，但绝不是一个完美的主题。因为他基本没有展现我们博客的内容，也没有漂亮的外观。展现博客的内容要靠 template tags（例如 bloginfo()）来完成，改变外观要靠 style.css 来完成。这也是制作主题的关键，最能展示个人创意的地方。接下来，如果您愿意把自己制作的主题 release 出来的话，就可以打包成.zip 文件，然后提交到 WordPress。

恩，现在是不是对 WordPress 主题制作有个基本的了解了，接下来我们会详细的讲解 template tags 的相关的知识。

## 第三讲 制作博客的 header

通过[上一讲](#)的说明大家知道了 [WordPress 主题](#)最主要的两个文件是 `style.css` 和 `index.php`，他们是最基本的，但是他们显然是不完美的，虽然你可以把所有的功能都写到 `index.php` 中，但这不符合 WordPress 主题制作的规范，这一讲我们将学习如何按照规范去制作自己的主题。



WordPress 的页面结构

从上面的图示我们可以看出一个基本的 WordPress 页面是由四部分构成的 (`header.php`, `index.php`, `footer.php`, `sidebar.php`)，他们在 [WordPress 博客](#)中负责不同的页面部分。我们将在下面的课程中讲解这四个文件的制作方法，今天我们就先来看看怎样来制作 `header.php` 文件。

(1) 首先我们在 `index.php` 中写入如下几个简单的函数，这些函数暂时不做讲解，等讲到 `index.php` 文件时我们会进行说明的。

代码如下：

```
<?php get_header()?>
```

```
<?php get_footer()?>
```

(2) 然后在 `wpc-tomheng` 目录下建立一个 `header.php` 文件。

写入如下代码：

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title> <?php bloginfo('name'); ?></title>

<linkrel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" media="screen"
/>

</head>

<body>

<div >

<h1><a href="<?php echo get_option('home'); ?>/"><?php bloginfo('name'); ?></a></h1>

<div class="description"><?php bloginfo('description'); ?></div>

</div>

</body>

```

(3) 打开浏览器查看主题是不是可以看到显示出博客的名字还有描述啦。

恩，这就是模版函数的功劳了，下面我们就看看这个是如何实现的。

说明：

### 一、bloginfo()模版函数

Bloginfo()就是个模版函数，通过它我们可以调用你博客相关的信息，我们在 header.php 中用到了四次这个函数，取出了三个主要的数据：博客的名字 bloginfo('name')、样式文件 bloginfo('stylesheet\_url')、博客描述 bloginfo('description')。这里需要特别指出的有两点：

- 1) Bloginfo()函数具有输出功能，即他不是返回数据，而是直接把数据输出到浏览器。
- 2) Boginfo()输出不同的数据通过参数配置实现。

Bloginfo()函数的常用配置参数说明



参数名	说明	返回值举例
html_type	文档类型	Text/html
charset	网页编码	Utf8
name	博客的名字	趣味互联网
stylesheet_url	Style.css 的 url 地址	<a href="http://blog.webfuns.cn/wp-content/themes/inove/style.css">http://blog.webfuns.cn/wp-content/themes/inove/style.css</a>
comments_rss2_url	评论的 rss 订阅地址	<a href="http://blog.webfuns.cn/comments/feed">http://blog.webfuns.cn/comments/feed</a>
description	副标题	webfuns
admin_email	管理员的邮箱	admin@webfuns.cn
version	Wordpress 的版本	2.8
language	当地的语言	Zh-cn

## 二、get\_option()函数

我们在这里用这个函数得到了博客链接的地址，这个函数需要注意就一点他返回的是一个 url 地址变量，所以如果要在页面上显示那么需要在他的前面加上 echo 函数。

当然我们还应该加上一些样式，但是这超出了我们教程的范围，我在这里只是向大家解释如何使用这些模版函数和主题制作的基本方法，并不会去叫大家如何设计主题，掌握了这些知识再加上丰富的创意才能实现漂亮的主题，这要依赖于你的 css 知识基础和设计能力而这些不包含在我们教程的内容之中（也许以后 wpc 会提供相关的教程，请密切关注）

（2）打开浏览器查看主题是不是可以看到显示出博客的名字还有描述啦。恩，这就是模版函数的功劳了，下面我们就看看这个是如何实现的。

说明：

### 一、bloginfo()模版函数

Bloginfo()就是个模版函数，通过它我们可以调用你博客相关的信息，我们在 header.php 中用到了四次这个函数，取出了三个主要的数据：博客的名字 (bloginfo('name'))、样式文件(bloginfo('stylesheet\_url'))、博客描述bloginfo('description')。这里需要特别指出的有两点：

1)bloginfo()函数具有输出功能，即他不是返回数据，而是直接把数据输出到浏览器、

2) Bloginfo()输出不同的数据通过参数配置实现。

## Bloginfo ()函数的常用配置参数说明

参数名	说明	返回值举例
html_type	文档类型	Text/html
charset	网页编码	Utf8
name	博客的名字	趣味互联网
stylesheet_url	Style.css 的 url 地址	http://blog.webfuns.cn/wp-content/themes/inove/style.css
comments_rss2_url	评论的 rss 订阅地址	http://blog.webfuns.cn/comments/feed
description	副标题	webfuns
admin_email	管理员的邮箱	admin@webfuns.cn
version	Wordpress 的版本	2.8
language	当地的语言	Zh-cn

## 二、get\_option () 函数

我们在这里用这个函数得到了博客链接的地址，这个函数需要注意就一点他返回的是一个 url 地址变量，所以如果要在页面上显示那么需要在他前面加上 echo 函数。

当然我们还应该加上一些样式，但是这超出了我们教程的范围，我们只是向大家解释如何使用这些模版函数和主题制作的基本方法，并不会去叫大家如何设计主题，掌握了这些知识再加上丰富的创意才能实现漂亮的主题，这要依赖于你的 css 知识基础和设计能力而这些不包含在我们教程的内容之中（也许以后 wpc 会提供相关的教程，请密切关注 wpc 的动向）

## 第四讲 制作博客的 footer.php

上一讲，我们讲了如何制作 WordPress 主题的 header 部分，到目前为止，我们的主题目录中应该有 style.css, header.php, index.php 这些基础文件，今天我们讲一下 footer.php 的制作。

footer.php 就是负责页面的底部部分，这个部分一般包含的内容主要有版权信息、统计代码、备案信息（在国内）等。现在我们在 wpc-tomheng 目录在建立一个 footer.php 文件。在里面写入如下代码：

```
<div id="footer">

<div ><a href="<?php bloginfo('url'); ?>"><?php bloginfo('name');?>Powered by </a><a href="">WordPress</a></div>

<div >

<div class="tongji">在这里可以加入你的统计代码</div>

<div class="beian">这里是您的备案信息</div>

<div ><?php echo get_num_queries();?>quers in <?php echo timer_stop() ;?>seconds</div>

</div>

<?php wp_footer(); ?>

</body>

</html>
```

现在打开浏览器应该可以看到相关的信息。确实很简单吧，*it is easy ,it is fun.*

Bloginfo()函数就不用说了，如果你不是很明白的话，那么请查看我们上一讲的内容。

### （1）get\_num\_queries()函数

这个函数（不需要任何的参数）的作用就是得到从 WordPress 初始化到这个函数调用为止，总共进行了多少次 sql 查询。

### （2）timer\_stop()函数

这个就是取得当前程序的运行时间，可以大体衡量一下 WordPress 查询的时间。

这个函数定义在：/wordpress/wp-settings.php -> line 162 有兴趣的朋友可以查看一下。

### （3）wp\_footer()函数

Wp\_footer()是 WordPress 的一个 hook 函数，用于插件开发中挂在函数。加上这种函数是自己的主题支持插件开发，同时和更符合规范。

今天涉及的这几个函数比较简单没有复杂的参数选项，相对简单一下，这些函数只要知道他们的作用，具体用在哪里要看自己的设计啦。

## 第五讲 制作博客的 **index.php** 文件

经过前面几讲的学习，我们已经了解了 footer.php 、 header.php 、 style.css 的基本知识。今天要讲一下最重要的一个文件 index.php。

在 index.php 中主要的就是取出文章相关的信息，这也是页面里的主体信息。我们在 header.php 里有提到过这个文件，还写入了两行代码。接下来我们重新写入如下代码，并且讲解这里面用到的模版函数。

```
<?php get_header(); ?>

<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

<div class="post" id="post-<?php the_ID(); ?>">

<h2><a class="title" href="<?php the_permalink() ?>" rel="bookmark"><?php
the_title(); ?></a></h2>

<div class="info">

<span class="date"><?php the_time() ?></span>

<?php if ($options['author']) : ?><span class="author"><?php
the_author_posts_link(); ?></span><?php endif; ?>

<?php edit_post_link('edit', '<span class="editpost">', '</span>'); ?>

<span class="comments"><?php comments_popup_link('no comment', '1
comment', '%
comments', 'Comments off'); ?></span>

<div class="fixed"></div>

</div>

<div class="content">

<?php the_content('read more'); ?>

</div>

<div class="under">

<span class="categories">Categories</span><span><?php the_category(' '); ?></span>

<span class="tags">tags: </span><span><?php the_tags(", ' ', '); ?></span>

</div></div>

<?php endwhile; else : ?>

<div class="errorbox">

没有任何文章

</div>
```

```
<?php endif; ?>
```

```
<div id="pagenavi">
```

```
<?php if(function_exists('wp_pagenavi')) : ?>
```

```
<?php wp_pagenavi() ?>
```

```
<?php else : ?>
```

```
<span class="newer"><?php previous_posts_link('上一页'); ?></span>
```

```
<span class="older"><?php next_posts_link('下一页'); ?></span>
```

```
<?php endif; ?>
```

```
</div>
```

```
<?php get_footer() ?>
```

请注意其中的引号，应该是英文半角的。

这些代码是不是看起来有些眼晕啊，这没关系，我们慢慢来分析。

### （1）**get\_header()**，**get\_footer()**函数

还记得我们前两讲里做的 header.php 和 footer.php 文件吗，这两个函数的作用就是把这两个文件引入进来，就相当于 php 中的 include 函数，这样我们才能在同一个网页中把 header 和 footer 都显示出来。这也可以看出来 index.php 的重要性，他统领着其他的文件。与这两个函数类似的还有 get\_sidebar()函数，这个函数就是包含 sidebar.php 文件，这个文件的内容我们将在后续的课程中讲解。

### （2）**have\_posts()**，**the\_post()**函数

Have\_posts()的作用就是判读博客中是不是有发表的文章，通常用于条件判断中，the\_post()的作用就是进行数据库查询，把与文章相关的数据查询出来，这样我们就能在后续的代码中使用相关函数将这些信息输出到页面上，一般的用法就是像上面那样：

```
<?php If(have_posts()):while(have_posts()):the_post();?>
```

这里是循环取出与文章相关的信息

```
<?php endwhile;?>
```

```
<?php else:?>
```

这里是如果没有文章的话，就显示这里的信息。

```
<?php endif;?>
```

需要注意的是这里的 if 和 while 后面用的是冒号不是分号，如果误写成分号就不能实现预想的功能啦。

(3) **the\_ID()**    **the\_permalink()**    **the\_title()**    **the\_time()**    **the\_content('read more')**

附注：这五个函数只能在 WordPress 的循环中使用，他们不是全局函数。

The\_ID()

作用是取得每次循环的编号，输出的结果 post-数字编号

The\_permalink()

作用是取得文章永久链接。这个函数不带参数表示取得当前循环中的文章的链接，也可以传入一个 id 编号这样就可以取得该 id 所对应的文章的链接。函数定义在：  
/wordpress/wp-includes/link-template.php -> line 9

The\_title()

取得文章的标题，这个函数的完整参数列表 the\_title(\$before = "", \$after = "", \$echo = true)

\$before 就是指文章标题前面要显示的内容，\$after 就是指文章标题后面说要显示的内容。  
\$echo 取值 true 表示显示文章标题，为 false 则返回文章的标题。

函数定义在： /latest/wordpress/wp-includes/post-template.php -> line 35

The\_time()

the\_time()用来获取并显示当前文章发布的时间,该函数后面可以跟控制日期或时间格式的参  
数，常用的参数形式如下：

如<?php the\_time('F j, Y'); ?>的调用形式显示效果为：六月13, 2009(英文状态下显示 June 13, 2009);

如<?php the\_time('g:i a'); ?>的调用形式显示效果为： 7:09 下午（英文状态下显示7:09 pm);

如<?php the\_time('G:i'); ?>的调用形式显示效果为： 19:09。

事实上,除了使用 the\_time()函数之外,WordPress 还提供了一个具有类似功能的 get\_the\_time() 函数。该函数除了不具有 the\_time()函数的显示功能之外，其余功能二者完全相同。使用 get\_the\_time()函数时，如欲将获取的时间显示出来，需要使用专用语句。

下面，我们籍此机会来简单了解一下 WordPress 中时间的格式。在 WordPress 中，通常用于控制时间格式的有以下字符：l, F, j, S, Y, G, g, i, a 等等，其详细意义如下：

l（小写 L）用来显示一周之中每一天的名称，比如星期六，或者在英文中显示 Saturday;

F 用来显示月份名称，比如六月，或者 June;

j 用来显示一月之中的某一天，比如13;

Y 用来以4位数字形式显示年份，使用 y 则以末两位数字显示年份，比如2009或09;

G, g, i, a 等四个字符通常组合使用，如前例子，有两种形式：

g:i a 以形如7:09 下午或7:09 pm 的形式显示时间;

G:i 以形如19:09的24小时进制形式显示时间。

S 通常紧跟在 j 后面, 表示是否在一月之中某天之后添加英文后缀 (st, nd, th 等)。

The\_content()

取出文章的内容, 这个可以带一个参数,the\_content(“more”),作用是如果在文章中有<!--more-->时, 就自动截断显示 more 连接, 可以查看详细信息。

#### (4) the\_category()、the\_tags()函数

the\_category() 取得文章所属的目录列表, 可以这样使用 the\_category('|'),如果有 WordPress WordPress 主题 WordPress 插件这三个目录那么显示出来是这样的:

WordPress|WordPress 主题|WordPress 插件, ‘|’就是相邻目录之间的分隔符, 这个可以根据自己的需要来使用不同的符号把他们分隔开来, 默认情况下是使用空格来分隔的。

the\_tags() 取得文章所属的标签。

the\_tags( \$before = ‘Tags: ‘, \$sep = ‘, ‘, \$after = ” ) \$before 在标签前面显示的内容, \$sep 分隔符, \$after 标签后显示的内容。

## 第六讲 sidebar.php 制作流程

上一讲我们学习了 [index.php 的制作](#)，这一讲比较重要一些，大家如果有不明白的地方，回头仔细看看我们教程的内容，一定要把 `index.php` 中的代码理解清楚。今天我们就来看看 `sidebar.php` 的制作。

首先我们需要在 `index.php` 中加入 `get_sidebar()` 函数来引入这个文件。打开 `index.php` 文件，在 `get_footer()` 函数前面加上 `get_sidebar()` 函数就可以啦。

接下来在 `sidebar.php` 中写入如下代码：

```
<div id="sidebar">

<ul>

<li>

<?php get_search_form(); ?>

</li>

</ul>

<ul role="navigation">

<?php wp_list_pages('title_li=<h2>' . 'Pages' . '</h2>' ); ?>


<li><h2>Archives</h2>

<ul>

<?php wp_get_archives('type=monthly'); ?>

</ul>

</li>


<?php wp_list_categories('show_count=1&title_li=<h2>' . 'Categories' . '</h2>'); ?>

</ul>

<ul>

<?php if ( is_home() || is_page() ) { ?>

<li><h2>Meta</h2>

<ul>

<?php wp_register(); ?>

<li><?php wp_logout(); ?></li>
```



```
</ul>

</li>

<?php } ?>

</ul>

</div>
```

现在可以打开浏览器看看这些代码都是 干了些什么啦。下面我们来看看里面的函数的作用是什么、如何使用。

### (1) **get\_search\_form()**函数

这个函数的作用就是现实搜索框。这个函数调用后就可以直接显示一个具有完备功能的搜索框，我们不用写任何额外的代码。这就是 WordPress 的魅力，他给我们做了一下很有用的封装函数，这样就减轻了我们的开发负担，可以专注于主题的设计。

### (2) **wp\_list\_pages()** 、 **wp\_list\_categories()**函数

**wp\_list\_pages()**

该函数以页面名称的超链接形式显示 [WordPress](#) 博客内的所有页面，经常用来建立顶端导航页面，或用来修饰侧边栏。

形如 `wp_list_pages('title_li=&sort_column=menu_order&include=12,25,38,57&depth=1&');`用，各参数意义如下：

**title\_li=** 用来设置所有显示页面的一个总名称；后面没有参数值时，表示不显示名称；

**sort\_column=menu\_order** 用来设置页面的显示顺序，表示按照 WP 后台设置的各页面顺序显示，其他的常用顺序设置可能还包括 **post\_title**、**post\_date**、**ID** 等等；

**include=12,25,38,57** 表示只显示 ID 为这四个数值的四个页面；我们也可以使用 **exclude** 来排除相应 ID 的页面；

**depth=1** 表示只显示父页面，对所有子页面不予显示；其他数值还包括默认的0，表示显示所有页面（子页面有缩进）；-1显示所有页面（子页面无缩进）；等等。

此外，该函数可能会用到的属性还包括 **link\_before** 和 **link\_after**，用于设置显示的页面链接前后的字符。

**wp\_list\_categories()**

**wp\_list\_categories()**函数用来获取博客文章的分类信息，并可以通过设置适当的函数参数，将其显示出来。该函数的参数和 **wp\_get\_archives()**函数类似，都需要使用&连接，放在单引号 (') 中以字符串方式传递。形如 `wp_get_archives('orderby=name&order=ASC&show_count=1&use_desc_for_title=1&feed=订阅&exclude=2,5& number=10')`

各参数的意义如下：

`orderby=name` 表示按照分类名称的字母先后顺序显示分类信息，可以将 `name` 换为 `ID` 等；  
`order=ASC` 表示按照分类名称的字母的升序显示分类信息，将 `ASC` 改为 `DESC` 表示按降序；

`show_count=1` 在每个分类名称后面显示属于该分类的文章数；

`use_desc_for_title=1` 使用该分类的描述信息为每个分类名称超链接添加一个 `title` 属性；

`feed=` 订阅：在每个分类信息旁边添加一个名为“订阅”的超链接，提供该分类的 RSS 订阅；

`exclude=2,5` 在显示的分类中去除 ID 为 2 和 5 的分类；也可以用 `include=2,5` 表示只显示 ID 为 2 和 5 的分类；

`number=10` 表示只显示前面的 10 个分类。

### (3) `is_home()` 、 `is_page()` 函数

`is_home()` 用以判断当前显示的博客页面是否是首页，返回的是一个 `Bool` 值。如果是在首页，则返回 `TRUE`；否则返回 `FALSE`。

该函数常用来控制博客侧边栏的显示方式，经常使用如下代码段：

```
<?php
if ( is_home() ) {
//此为在博客首页应该显示的内容
} else {
//此为非博客首页应该显示的内容
}
?>
```

`is_page()` 函数判断当前显示的内容是否是博客的独立页面（`page`），就是在后台建立的页面，它也返回一个 `Bool` 值。我们可以在模板中通过该函数判断当前是否是一个独立页面，从而决定是否当前显示的文章显示发布时间等等。

同类的函数还有：

`is_paged()` `is_single()` `is_tag()` `is_category()` `is_archive()` 等等

### (4) `wp_register()` 、 `wp_loginout()` 函数

`Wp_register()` 获得登录后台的链接，`wp_loginout()` 获得退出登录的链接。

到现在，一个基本的 [wordpress 主题](#) 就已经成形了。这套主题教程也就告一段落了，希望大家会从中有所收获。