

WordPress 主题制作教程

小雯子@WordPress 啦！写作

作者: 小雯子@WordPress 啦!
制作者: 大漠孤狼@站长百科
制作时间: 2009年2月2日
制作网站: 站长百科 www.zzbaike.com
版权: WordPress 啦！及作者

WordPress 主题制作电子书使用指南

本文档是 WordPress 主题制作的入门指导，详细分析了 WordPress 主题的每个文件，并对常用的函数使用参数做了详细的介绍。只要您具有基本的 html 及 css 知识，通过学习本文档就可以掌握 WordPress 主题的制作了

为了方便大家阅读，制作了详细的目录及书签。对于 PDF 电子书来说，我认为书签使用起来更方便一些，希望大家能够习惯电子书的阅读方式。

本电子书由WordPress啦！的小雯子写作，我调整了一下版面并制作成本电子书，获取WordPress的更多信息请大家访问WordPress啦（www.wordpress.la），上面有WordPress使用手册，评论，主题，以及关于WordPress的最新新闻。

如果您有资料想制作成电子书请联系我，我们会协助您制作和发布。
有问题或建议请到我的博客留言：www.83blog.com

本电子书的版权归 WordPress 啦及作者所有，可以自由复制，转载，但请勿将本电子书内容或其中的一部分私自用于出版物。如果需要用于商业用途请联系我们。如果引用部分内容，请注明作者及来源。

联系方式：webmaster@wordpress.la 或到网站留言。

一. WORDPRESS 模板文件	2
二. 模板文件详细	3
1、HEADER.PHP	3
2、INDEX.PHP	4
3、SIDEBAR.PHP	8
4、FOOTER.PHP	14
5、FUNCATIONS.PHP	14
6、ARCHIVES.PHP	15
7、SEARCHFORM.PHP 和 COMMENTS.PHP	16
9、把 INDEX.PHP 另存为 SEARCH.PHP、PAGE.PHP、SINGLE.PHP	17
三. 精确控制你的模板	18
1、用图片做 LOGO	18
2、关于 SIDEBAR (边栏的样式)	18

左侧有书签导航, 点击书签就可以进入对应页面, 很方便。为了方便大家使用, 还专门制作了函数的书签, 点击可以进入查看函数的用法及参数。

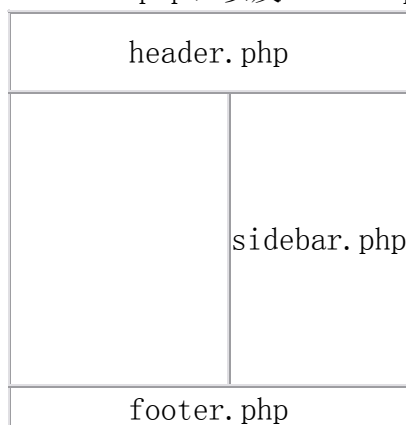
一. WordPress 模板文件

wordpress 里面有很多 php 文件，但是你使用了就会发现文件很简单，只要把 wordpress 的一般格局和功能搞清楚了，文件你也就知道个差不多了。

wordpress 的格局其实就像在做网页的时候一样，分为几个大类：

1. 头部文件：header.php
2. 主体内容部分： index.php
3. 页面文件： page.php
4. 单篇日志页面： single.php
5. 边栏文件： sidebar.php
6. 页脚文件： footer.php

index.php\page.php\single.php 是主体文件，里面包含着 header.php、sidebar.php、以及 footer.php



主体部分文件由于所在页面不同，会导入不同的页面文件，一般情况下 index.php 是首页文件。

上面的是按照格局所得出的一些 wordpress 文件，还有按照功能得到的一些文件：

1. 404 错误： 404.php
2. 文章存档： archives.php
3. 评论功能： comments.php
4. 自定义功能： functions.php
5. 搜索结果页面： search.php
6. 搜索表单： searchform.php

这些文件齐了以后，在加上 css 样式文件 style.css 以及图片文件夹 images，博客也就有了大致的雏形了。

二. 模板文件详细

1、header.php

包含了模板的基本信息, 有 meta 和 title 以及 link 的信息, 还有显示博客的标题与副标题, 放置导航栏等

```
<div id="header"> (头部)
```

```
<h1><a href=" <?php  
bloginfo(' url' );?>" ><?phpbloginfo(' name' );?></a></h1> (标题)
```

```
<?php bloginfo(' description' );?> (副标题)
```

```
<div id="nav"> (页面导航)
```

```
<?php wp_list_pages(' title_li=&depth=1' ); ?> (以有序列表的形式获得页面的链接)
```

```
</div>
```

```
</div>
```

附: header.php 页面里经常用到的函数及其参数

`<?php get_bloginfo (' '); ?>` : 将由 wordpress 函数自动生成的信息传回博客, 然后再在其他地方使用你的 PHP 代码

部分常用参数:

default: 默认

name: 名称

description: 说明

url、home、siteurl: 网址

charset: 网页的编码方式

stylesheet_url: css 文件地址

version: 博客所使用的 wordpress 的版本

例: `<?php echo 'Your Blog Tagline is: ' . get_bloginfo('description'); ?>`

`<?php bloginfo(' '); ?>` : 显示博客的信息 (参数同 `get_bloginfo`)

`<?php wp_list_pages(' '); ?>`: 列表显示页面链接

部分常用参数:

post_title: 按标题字母顺序排列 (默认)

post_date: 按创建时间排序

post_modified: 按最后修改时间排序

ID: 按照 id 排序

post_author: 按照作者 id 排序

2、 index.php

这是网页的主体内容文件, 也是站点的首页文件

`<?php get_header(); ?>` (获得头部文件)

`<div id="content" class="narrowcolumn">` (主体内容)

`<?php if (have_posts()) : ?>` (检查是否有 post 日志)

`<?php while (have_posts()) : the_post(); ?>` (当有 post 的时候 post...)

`<div class="post" id="post-<?php the_ID(); ?>">` (页面内容并给页面内容赋予 id)

`<h2><a href="<?php the_permalink() ?>" rel="bookmark" title="Permanent Link to <?php the_title_attribute(); ?>"><?php the_title(); ?></h2>`
(获得页面标题及链接)

`<small><?php the_time('F jS, Y') ?><!-- by <?php the_author() ?> --></small>` (页面的日期及作者)

`<div class="entry"><?php the_content(' Read the rest of this entry »'); ?></div>` (获得日志的内容)

```
<p class="postmetadata"><?php the_tags('Tags: ', ' ', ' ', ' <br />'); ?>
Posted in <?php the_category(' ', ' ') ?> | <?php edit_post_link('Edit', ' ',
' | '); ?> <?php comments_popup_link('No Comments &#187;', '1 Comment
&#187;', '% Comments &#187;'); ?></p> (获得日志的 tag 标签、分类、编辑
以及留言的信息及链接)
```

```
</div>
```

```
<?php endwhile; ?> (while 循环结束)
```

```
<div class="navigation"> (页面链接)
```

```
<div class="alignleft"><?php next_posts_link('&laquo; Older
Entries') ?></div>
```

```
<div class="alignright"><?php previous_posts_link('Newer Entries
&raquo;') ?></div>
```

```
</div>
```

```
<?php else : ?> (否则)
```

```
<h2 class="center">Not Found</h2>
```

```
<p class="center">Sorry, but you are looking for something that isn't
here.</p>
```

```
<?php include (TEMPLATEPATH . "/searchform.php"); ?>
```

```
<?php endif; ?> (if 判断结束)
```

```
</div>
```

```
<?php get_sidebar(); ?> (获得 sidebar 文件)
```

```
<?php get_footer(); ?> (获得 footer 文件)
```

附:

index.php\page.php\single.php 页面里经常用到的函数

<?php the_ID(); ?>: 特定内容页(Post)自动生成的 ID (无参数)

<?php the_permalink() ?>: 得到内容页(Post) URL (无参数)

`<?php the_title(); ?>`: 显示或返回内容页 (Post) 标题

用法: `<?php the_title('before', 'after', display); ?>`

before: (字符串) 标题前面的文字, 默认情况下不显示,

after: (字符串) 标题前面的文字, 默认情况下不显示

display: (布尔) 当布尔值为真时, 显示标题 (默认), 当布尔值为假时, 得到标题的值但不显示

`<?php the_time('F jS, Y') ?>`: 日志发表的日期 格式: 月日年

时间的格式有很多种, 写在下面, 供参考

年-月-日

(1) 表示年可以用大写的 Y 和小写 y; (大写 Y 表示年四位数字, 而小写 y 表示年的两位数字)

(2) 表示月可以用大写 F、大写 M、小写 m 和小写 n(小写 m 表示月份的数字(带前导), 而小写 n 则表示不带前导的月份数字。大写 M 表示月份的 3 个缩写字符, 而小写 m 则表示月份的数字(带前导 0); 大写 M 表示月份的 3 个缩写字符, 而大写 F 表示月份的英文全写)

(3) 表示日可以用小写 d 和小写 j, 大写 S 表示日期的后缀。(没有大写的 J, 只有小写 j 表示月份的日期, 无前导 o; 若需要月份带前导则使用小写 d; 大写 S 表示日期的后缀, 比如 “st”、“nd”、“rd” 和 “th”, 具体看日期数字为何)

时:分:秒

(1) i 表示分, s 表示秒

(2) 小写 g 表示 12 小时制, 无前导 0, 而小写 h 则表示有前导 0 的 12 小时制。

(3) 当使用 12 小时制时需要表明上下午, 小写 a 表示小写的 “am” 和 “pm”, 大写 A 表示大写的 “AM” 和 “PM”。

(4) 大写 G 表示 24 小时制的小时数, 但是不带前导的; 使用大写的 H 表示带前导的 24 小时制小时数

注: 默认情况下, PHP 解释显示的时间为 “格林威治标准时间”, 与我们本地的时间相差 8 个小时。

`<?php the_author() ?>`: 显示日志的作者 (无参数)

`<?php the_content('Read the rest of this entry »'); ?>`: 日志的内容

用法: `<?php the_content('more_link_text', strip_teaser, ''); ?>`

more_link_text: (字符串), 当日志有 more 标记时, 链接到日志页面

strip_teaser: (布尔), 日志“更多”链接的隐藏 (true) 或显示 (false) (默认)

`<?php the_tags('Tags: ', ', ', ', '
'); ?>`: 日志的 tag 标签

用法: `<?php the_tags('before', 'separator', 'after'); ?>`

before: (字符串) 用在标签的前面, 默认为“tags: ”

separator: (字符串) 分隔符, 默认情况下不显示

after: (字符串) 用在标签的后面, 默认情况下不显示

`<?php the_category(', ') ?>`: 日志所属的分类

用法: `<?php the_category('separator', 'parents'); ?>`

separator: (字符串) 分隔符, 默认情况下是无序列表

parents: (字符串) 显示分类的级别

`<?php edit_post_link('Edit', '', '| '); ?>`: 当用户已登录并具有权限时, 所显示的日志编辑链接

`<?php comments_popup_link('No Comments »', '1 Comment »', '% Comments »'); ?>`: 日志的评论情况

`<?php next_posts_link('« Older Entries') ?>`: 较早日志页面的链接

`<?php previous_posts_link('Newer Entries »') ?>`: 较新日志页面的链接

3、sidebar.php

侧边栏文件

```
<div id="sidebar">

<ul>

<?php if ( !function_exists('dynamic_sidebar')
|| !dynamic_sidebar() ) : ?> (当检测到当前的 WordPress 是支持动态侧边栏
时, 则边栏 )

<li id="pages">

<h2>
<?php _e('pages');?>
</h2>

<ul>
<?php wp_list_pages('sort_column=menu_order&depth=1&title_li='); ?>
</ul>

</li>

<li id="meta">

<h2>
<?php _e('Meta'); ?>
</h2>

<ul>

<?php wp_register(); ?>
<li>
<?php wp_loginout(); ?>
</li>
<?php wp_meta(); ?>

</ul>

</li>

<?php endif; ?> (if 判断结束)

</ul>
```

</div>

附: sidebar.php 经常会运用到的函数

<?php wp_register();?>: 显示注或网站管理链接

用法: <?php wp_register ('before' , 'after'); ?>

before: (字符串) 默认为

after: (字符串) 默认为

<?php wp_logout();?>: 显示登录/注销链接 (无参数)

<?php wp_meta():?>: 显示自定义字段, 不过貌似从来没用上

<?php wp_get_archives('type=monthly');?>: 按月显示内容存档

用法: <?php wp_get_archives('type='&'limit='&'format='&'before='&'after='&'show_post_count='&')

参数及值:

type : (字符串) 显示档案的方法

- yearly : 年
- monthly (默认) : 月
- daily : 日
- weekly : 周

limit : (字符串) 显示文章数目的限制, 默认情况下无限制

format: (字符串) 文档清单的格式

html: 添加 HTML 列表 () 标签 (默认)

option: 添加选择 (<select>) 或下拉式选项 (<option>) 标记

link: 添加链接 (<link>) 标记

custom: 自定义列表使用之前和之后的字符串。

before: (字符串) 当有 format 标记时用在 format 前面

after: (字符串) 当有 format 标记时用在 format 后面

show_post_count: (布尔) 是否显示文档里面日志的数目, 1 显示 0 不显示 (默认)

<?php wp_tag_cloud(''):?>: 显示 tag 标签

用法:

```
<?php wp_tag_cloud('smallest=,
largest=,unit=,number=,format=,orderby=name,order=ASC,exclude=,include='); ?>
```

参数:

smallest : (整数) 文字的最小尺寸 (unit 给予单位)

largest: (整数) 文字的最大尺寸 (unit 给予加单位)

unit : (字符串) 给 smallest 和 largest 添加的单位

number: (整数) 显示的 tag 的数目

format : (字符串) 标签云显示的格式

值:

- flat: 由空格分开 (默认)
- list: 以有序列表排列
- array: 标签被放在一个数组里面, 并且函数将标签云返回为一个数组供 php 调用。注意: 数组是返回, 而不显示

orderby: (字符串) 标签的排列的顺序

值:

- name: 名字 (默认)
- count: 数字

order : (字符串) 排序的次序 (要大写才有效)

值:

- ASC: 升序 (默认)
- DESC: 降序
- RAND: 随机

exclude: (字符串) 排除某个特定的标签

include: (字符串) 只包含某个特定的标签

```
<?php $myposts=get_posts('numberposts=10$offset=1');foreach($myposts as $post):?><li><a href="<?php the_permalink();?>"><?php the_title();?></a></li><?php endforeach;?>
```

: 显示从第一篇开始取 10 篇日志的列表

<?php get_posts(' '); ?>: 得到日志的信息

用法: <?php get_posts('numberposts=, offset=, category=, category_name=, tag=, orderby= ,……'); ?>

参数及值:

numberposts: (整数) 返回日志的数目, 默认值: 5

offset: (整数) 最新文章的开始点, 默认值: 0

category: (整数) 只显示这一类别的文章 (根据 ID), 默认值: 无

category_name: (字符串) 只显示这一类别的文章 (根据名称或缩写), 默认值: 无

tag: (字符串) 只显示这一标签的文章, 若有多个标签, 用逗号隔开, 则一个不显示, 应用空格隔开

orderby: (字符串) 文章的排列顺序 (默认: post_date)

值:

- author: 作者 id
- category: 分类 id
- content: 内容 id
- date: 日期 id
- ID: 文章本身的 id
- menu_order: 按菜单命令 (只应用于网页 page)
- mime_type: 按 MIME 类型 (只应用于带附件的)
- modified: 最后修改的日期
- name: 按缩略名
- parent: 按父文章的 id
- password: 按密码
- rand: 随机
- status: 按状态
- title: 按标题
- type: 按类型

order: (字符串) 排序的次序 (要大写才有效)

值:

- ASC: 升序
- DESC: 降序 (默认)

include: (字符串) 显示拥有某些 ID 的文章, 多个以逗号、/、空格隔开, 默认值: 无

exclude: (字符串) 排除拥有某些 ID 的文章, 多个以逗号、/、空格隔开, 默认值: 无

meta_key 和 meta_value: (字符串) 只显示那些有特定键值对的自定义域, (如果要用的话, 两个值一起), 默认值: 无

post_type: (字符串) 文章的类型

值:

post (默认)

- page
- attachment
- any

post_status: (字符串) 查看文章的特殊状态

值:

- publish (默认)
- private
- draft
- future
- inherit

post_parent: (整数) 只显示这个拥有这个 id 的文章的子文章, 默认值: 无

nopaging: (布尔) 启用或禁用分页, 默认值: 无

<?php

```
wp_list_categories('sort_column=&optioncount=&hierarchical=&……');?
```

>: 按照字符顺序显示分类列表

参数:

show_option_all : (字符串) 当 style 设置为 list 的时候, 会有一个值来显示到所有分类的链接, 默认是不显示所有链接

orderby: (字符串) 排序的依据, 默认值: name

值:

- ID
- name (默认)
- count

order: ASC (升序) 或 DESC (降序)

show_last_update: 是 (真) 否 (假) 显示最后修改时间, 1 为真, 0 (默认) 为假

style: (字符串) 显示分类的方式, 值: list (默认); none

show_count: (布尔) 是 (真) 否 (假) 显示日志的数目, 1 为真, 0 (默认) 为假

hide_empty: (布尔) 是 (真) 否 (假) 隐藏无日志的分类, 1 (默认) 为真, 0 为假

use_desc_for_title: (布尔) 是 (真) 否 (假) 给标题设置分类信息, 1 (默认) 为真, 0 为假

child_of: (整数) 只显示这个分类的子分类, 没有默认值

exclude: 除了

include: 只包括

hierarchical: (布尔) 是 (真) 否 (假) 让子分类显示时缩进, 1 (默认) 为真, 0 为假

number: (整数) 设置显示的数目, 默认无限制

echo: (布尔) 得到的结果是 (真) 否 (假) 存在一个变量中, 1 (默认) 为真, 0 为假

depth: (整数) 显示分类的深度, 0 (默认) 为无限制

```
<?php  
wp_list_bookmarks(' categorize=&title_after=&title_before=&……');?>  
显示友情链接
```

```
<?php wp_list_pages(' sort_column=menu_order&depth=3&title_li=' ); ?>  
显示深度到 3 的页面链接
```

```
<?php include(TEMPLATEPATH. '/searchform.php' ); ?>: 调用 searchform  
的文件生成搜索栏
```

```
<?php get_calendar(); ?>: 调用 wordpress 自带的日历
```

4、footer.php

脚部文件，一般就用来申明版权

```
<div id="footer"><?php bloginfo('name'); ?> is proudly powered by<a  
href="http://wordpress.org/">WordPress</a>, designed by 作者</div>
```

5、funcations.php

自定义函数功能

一般情况下，funcations.php 里面定义的是支持 widget 的 wp 模板的 sidebar 自动生成的 widget 选项的。

```
<?php  
  
if ( function_exists('register_sidebar') )  
  
register_sidebar(array(  
  
'before_widget' => '<li id="%1$s" class="widget %2$s">',  
'after_widget' => '</li>',  
'before_title' => '<h2 class="widgettitle">',  
'after_title' => '</h2>',  
  
));?>
```

这是用来说明当检测到当前模板支持 widget 时, wordpress 生成的 widget 各选项会有一个 li 标签围绕, 而自动生成的各标题则自动围绕一个 h2 标签。

funcations.php 可以用来定义各种需要的函数, sidebar 只是一个小例子。

6、archives.php

分类功能

分类页面就是 index.php 文件的 `<?php if (have_posts()) : ?><?php while (have_posts()) : the_post(); ?>`中间添加一段分类的代码

```
<?php if (have_posts()) : ?>
```

```
<?php $post = $posts[0]; // Hack. Set $post so that the_date() works. ?>
```

```
<?php /* If this is a category archive */ if (is_category()) { ?>
```

```
<h2 class="pagetitle">Archive for the &#8216;<?php  
single_cat_title(); ?>&#8217; Category</h2> (根据 cate 分类)
```

```
<?php /* If this is a tag archive */ } elseif( is_tag() ) { ?>
```

```
<h2 class="pagetitle">Posts Tagged &#8216;<?php  
single_tag_title(); ?>&#8217;</h2> (根据 tag 分类)
```

```
<?php /* If this is a daily archive */ } elseif (is_day()) { ?>
```

```
<h2 class="pagetitle">Archive for <?php the_time('F jS, Y'); ?></h2>  
(根据 day 分类)
```

```
<?php /* If this is a monthly archive */ } elseif (is_month()) { ?>
```

```
<h2 class="pagetitle">Archive for <?php the_time('F, Y'); ?></h2> (根  
据 month 分类)
```

```
<?php /* If this is a yearly archive */ } elseif (is_year()) { ?>
```

```
<h2 class="pagetitle">Archive for <?php the_time('Y'); ?></h2> (根据  
year 分类)
```

```
<?php /* If this is an author archive */ } elseif (is_author()) { ?>
```

```
<h2 class="pagetitle">Author Archive</h2> (根据 author 分类)
```

```
<?php /* If this is a paged archive */ } elseif (isset($_GET['paged'])  
&& !empty($_GET['paged'])) { ?>
```

```
<h2 class="pagetitle">Blog Archives</h2> (根据 page 分类)
```

```
<?php } ?>
```

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

7、searchform.php 和 comments.php

搜索表单和评论功能

搜索表单和评论功能一般情况下大家可以直接作用默认模板里面的那两个文件，因为一般情况下这两个文件是不需要改动的。

而调用 cmments.php 则是用这个函数：<?php comments_template();?>：调用留言板函数

8、style.css

样式表文件

在文件的头部写上版权信息，这个很重要的，关系产权的问题，所以一定要写清楚

```
/*
```

```
Theme Name: 模版的标题
```

```
Theme URI: 模版的地址
```

```
Description: 模板的副标题
```

```
Author: 作者名字
```

```
Author URI: 作者链接地址
```

```
*/
```

然后在下面就是你所定义的 css 文件啦

9、把 index.php 另存为 search.php、page.php、single.php,

若是想在哪个页面开启留言功能的话，只要在页面中<?php comments_template(); ?>这段代码，就能够在这些页面留言了。

三. 精确控制你的模板

当懂了上面的那些文件以及能做出符合标准的 css 样式出来, 那么一个简单的 wordpress 模板做出来就不是问题了, 而如果要使模板更精美, 则需要更精确的来控制这些函数、使用这些函数。

1、用图片做 logo

大部分时候, 一个美观的页面的 logo 有着各种效果, 而这是简单的字符代替不了的, 所以我们就用到图片, 当然, 在 seo 的角度, 这个不太可取, 所以用漂亮的图片还是单调的字符, 就看作者的了, 我这里只是提供一种方法。

```
<div id="header">
<div id="headerimg">
<a href="<?php echo get_option('home'); ?>/"><div id="logo"></div></a>
</div>
```

用这段代码来代替

```
<div id="header">
<h1><a href=" <?php bloginfo('url') ;?>" ><?php
bloginfo(' name' );?></a></h1>
```

其实就是把 h1 的标签去掉, 换成 div headerimg 来定位这个 a 标记, 然后把自动得到模板标题的行为改成一张 logo 图片, 对一些定做模板或由于设计原因图片只能用标题的设计者比较有用。

2、关于 sidebar (边栏的样式)

sidebar 的样式设置用一个字来形容就是“繁”, 由于 wordpress 是自动生成的边栏 widget, 它会自己给一些 widgets 加上标签, 一般就是 ul 和 li, 而有的标签有三个子级, 所以设置样式的时候会要设置到多级, 宽度问题一定要把握精确, 要不然很容易就出现浏览器兼容的问题, 这个就不 累述了, 到时候多注意一些就好了。只有有些个别的 widget 的样式和 wordpress 自动自成的不一样, 这样, 在我们设置标签的时候, 这几个 widget 就要特别的另外设置了 (如果你的模板的边样样式复杂的话), 它们是 text、search、tag, 这个具体怎么设置, 具体的问题到时具体对待吧。