

Wordpress学堂 <http://Wordpress.suyun.org/>

Wordpress 主题制作教程

本文档由 wordpress 教程网(<http://www.54ux.com/>)整理，文中所有下载文件均在附件中。

来自 (<http://www.ludou.org>)

WordPress 主题制作全过程 (一): 基础准备

前言：

我想大多数使用 WordPress 的朋友都喜欢去尝试新的主题，但是换来换去，总是找不到那么一款适合自己的，让人很郁闷。于是很多人萌生了修改现有主题或自己动手从头制作一个主题的想法，但是问题又来了，怎么做呢？毕竟制作主题是需要编写代码的，相信很多人看到代码就头大。露兜博客正在着手编写一个 WordPress 主题制作的系列教程，将从头到尾手把手教你如何制作一个完整的 WordPress 主题，希望对你有所帮助。开始之前先声明，这系列教程只能让你感性认识主题的制作过程，并不能让你完完全全地掌握 WordPress 主题的原理。

另外，如果你在学习过程中遇到任何问题，欢迎给我留言，留言框在每篇文章末尾。

本系列教程适合以下人群阅读：

- 懂一点点的 HTML、CSS
- 自己动手编写过代码
- 自己会设计网页界面，想了解怎么制作 WordPress 主题

本系列教程不适合以下人群阅读：

- 不打算动手编写代码，不想照着本教程一步一步地操作，只是想看看
- 没有任何网页设计知识
- 没用过 WordPress
- 非常熟悉 WordPress 主题制作过程

- 不喜欢我在此罗嗦

阅读完本系列教程，你将收获：

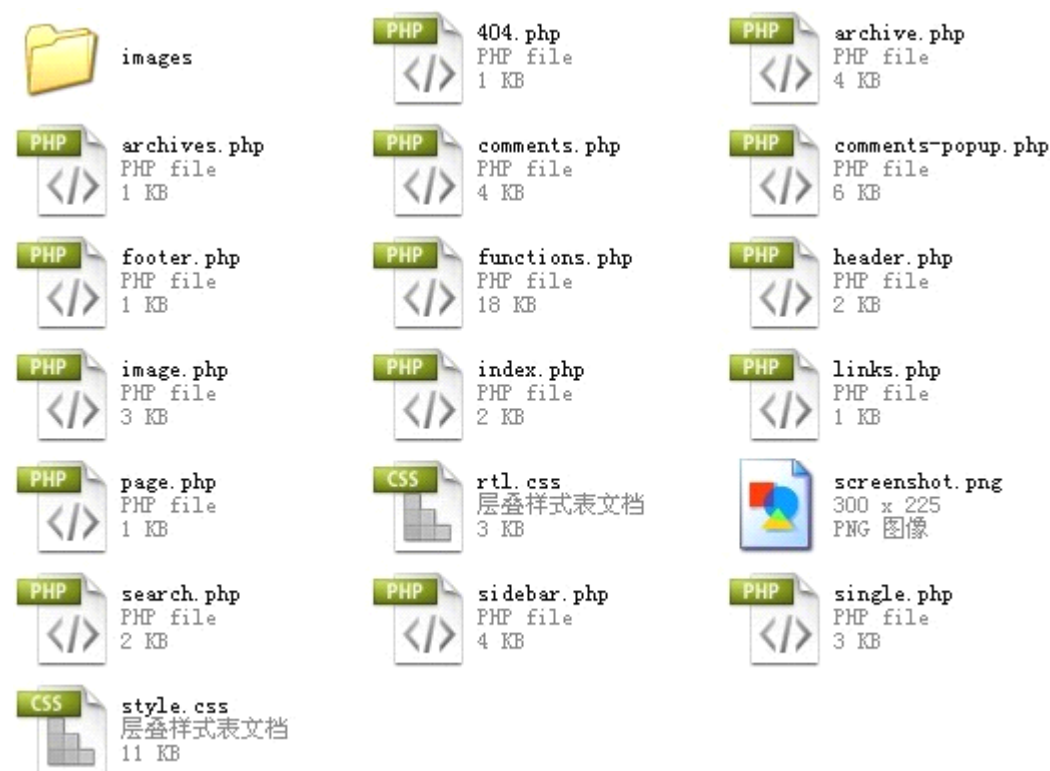
- 了解 WordPress 主题制作的过程
- 知道如何去修改一个 WordPress 主题，扩展主题的功能

开始制作主题之前，你必须配备以下工具：

- WordPress 本地运行环境（或者在线网站空间），并且已安装好 WordPress，作为测试博客
- 至少配备三种浏览器，Microsoft Internet Explorer、Mozilla Firefox、Google Chrome，版本暂时不限，用于测试网页是否显示正常
- 一个文本编辑器用于编写代码，Dreamweaver、UltraEdit、NotePad++、Vim 等

WordPress 主题制作全过程（二）：主题文件构成

在开始制作 WordPress 主题之前，首先得了解 WordPress 主题到底由哪些文件构成，你得清楚 WordPress 程序是怎样与主题文件连接的。以下是 WordPress 默认主题 default 文件夹下的所有模板文件：



看了上图，可能你还摸不着头脑，到底这些文件是干什么的。WordPress 的主题是用 PHP 编写的，而不是纯 HTML + CSS，所以模板文件的后缀名是.php，如果你想精通 WordPress 的主题制作，完美控制你的博客，最好要熟悉 PHP 编程。要是不会 PHP 编程怎么办？就做不了 WordPress 主题了吗？那也不是，至少看完本系列教程，你也能够掌握基本的 WordPress 主题制作方法。下面是 WordPress 主题文件层次结构，它会告诉你：当 WordPress 显示特定的页面类型时，会使用哪个模板文件呢？只有了解了以下主题层次结构，你才能知道你的 WordPress 主题到底需要写哪些文件。

怎么看下面的文件层次结构？

以主页为例，下面有2个文件 home.php 和 index.php，WordPress 程序会从你的主题文件夹中依次查找这两个文件，如果找到 home.php，则使用 home.php 作为博客首页模板，即使你的主题文件夹中有 index.php；如果 home.php 未找到，则使用 index.php 作为首页模板；如果 home.php 和 index.php 都找不到，你的主题将不会被 WordPress 识别，等于废物。

主页：

1 home.php

2 index.php

文章页：

3 single-{post_type}.php – 如果文章类型是 videos (即视频), WordPress 就会去查找 single-videos.php (WordPress 3.0及以上版本支持)

4 single.php

5 index.php

页面：

6 自定义模板 – 在 WordPress 后台创建页面的地方，右侧边栏可以选择页面的自定义模板

7 page-{slug}.php – 如果页面的缩略名是 news，WordPress 将会查找 page-news.php (WordPress 2.9及以上版本支持)

8 page-{id}.php – 如果页面 ID 是6，WordPress 将会查找 page-6.php

9 page.php

10 index.php

分类：

11 category-{slug}.php – 如果分类的缩略名为 news，WordPress 将会查找 category-news.php(WordPress 2.9及以上版本支持)

12 category-{id}.php -如果分类 ID 为6，WordPress 将会查找 category-6.php

13 category.php

14 archive.php

15 index.php

标签

16 tag-{slug}.php – 如果标签缩略名为 sometag , WordPress 将会查找 tag-sometag.php

17 tag-{id}.php – 如果标签 ID 为6 ,WordPress 将会查找 tag-6.php (WordPress 2.9 及以上版本支持)

18 tag.php

19 archive.php

20 index.php

作者：

21 author-{nickname}.php – 如果作者的昵称为 rami , WordPress 将会查找 author-rami.php (WordPress 3.0及以上版本支持)

22 author-{id}.php – 如果作者 ID 为 6 , WordPress 将会查找 author-6.php (WordPress 3.0及以上版本支持)

23 author.php

24 archive.php

25 index.php

日期页面：

26 date.php

27 archive.php

28 index.php

搜索结果：

29 search.php

30 index.php

404 (未找到)页面

31 404.php

32 index.php

附件页面：

33 MIME_type.php – 可以是任何 MIME 类型 (image.php, video.php, audio.php, application.php 或者其他).

34 attachment.php

35 single.php

36 index.php

详细说明，请看 WordPress 官方文档（如果你能看懂英文）：

http://codex.wordpress.org/Template_Hierarchy

WordPress 主题制作全过程（三）：HTML 静态模板制作

制作 WordPress 需要了解 PHP，但是不管怎样，你的博客页面都是一张网页，网页之所以能够在浏览器中显示出来，是因为最终传送给浏览器的是 HTML、CSS 和 JS 等终端代码，最终由这些代码控制浏览器的显示结果。制作 WordPress 主题也不例外，它们都是网页，网页最基本的是 HTML。要想制作 WordPress 主题，首先得设计主题界面，含蓄、简约、厚重、奔放……，风格完全随你，接着再将你的设计理念转化为.html，并能在浏览器中正常显示出来。

本系列教程不会教你如何去设计博客的界面，也不会教你 HTML 和 CSS 基础知识，而是假定你已经有 HTML 和 CSS 基础了，如果你不会 HTML，也不会 CSS，后面的东西恐怕对你来说就像天书了。

本系列教程将使用由 [tutsplus](#) 无偿提供的 html 模板 [Aurelius](#) 为例，来讲解 WordPress 主题的制作过程，其他模板的制作也类似。下面提供该 HTML 模板文件下载，我们将以此来练习如何将 HTML 静态页面与 PHP 代码结合，制作 WordPress 主题。下载后，里面的 html 文件都是可以直接双击打开的。**(文件已包含在附件中)**



可能这个模板并不适合每个人的口味，但这个用于我们的主题制作练习已经足够了。另外你可能会诧异，怎么是英文版的？放心，到了后面你就会知道，是否英文版都没多大关系。下面说说下载到的这些文件的用途：

- 首页：index.html
- 存档页：archive.html
- 页面：page.html
- 文章页：single.html
- 联系页：contact.html
- 无边栏页：full_width.html
- 图片：/images/
- 样式表：style.css
- 缩略图：screenshot.png

大多数博客所能看到的页面类型也基本就这些。你可能已经注意到，不管你用哪个

WordPress 主题，不同的文章页面除了标题和内容不同以外，其他东西基本都是一样，这就是所谓的模板。

WordPress 主题制作全过程（四）：小试牛刀

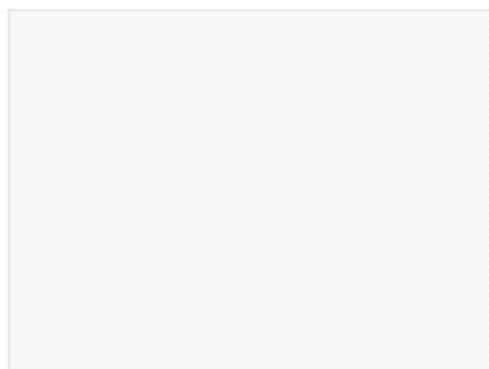
一个 WordPress 主题至少包含以下两个文件：

- style.css
- index.php

好，现在就开始我们的 WordPress 主题制作。在你的 WordPress 目录 wp-content\themes\ 下新建一个文件夹，命名为 Aurelius，将 [WordPress 主题制作全过程（三）：HTML 静态模板制作](#) 下载到的 style.css 和 index.html 放到该目录下，再将 index.html 重命名为 index.php，这样 wp-content\themes\Aurelius 目录下就有下面两个文件：

- style.css
- index.php

登陆你的 WordPress 博客管理后台 - 外观，里面是不是多了一个主题 "Aurelius" 呢？如下图所示：



Aurelius 1.0 作者为 [Ludou](#)

It's a theme create by Ludou

[启用](#) | [预览](#) | [删除](#)

该主题的所有文件存放在 `/themes/Aurelius` 。

标签： custom header, fixed width, two columns, widgets

启用该主题，打开你的博客主页，OH，页面一片混乱？怎么搞的？先继续看下文。

你可能会很好奇，这个主题的名称可不以改成我要的名字呢、主题作者能不能改成我的名字呢、为什么主题没有小图片预览，下面就教你怎样更改这些"主题信息"。用文本编辑器打开 Aurelius 目录下的 style.css，你可以按照下面的说明尝试在 style.css 文件最前面添加以下信息，再到 WordPress 管理后台"外观"栏目下，刷新一下看看有什么变化：

```
1  /*
2  Theme Name: 这里填主题名称
3  Theme URI: 这里填主题介绍的网址，没有就填你的博客网址
   吧
4  Description: 这里填主题的简短介绍
5  Version: 版本号
6  Author: 作者名
7  Author URI: 作者的网址
8  Tags: 标签，多个用半角逗号隔开
```

```
9 */
```

将 [WordPress 主题制作全过程\(三\):HTML 静态模板制作](#) 下载到的 screenshot.png 放到 Aurelius 目录下,再到 WordPress 管理后台"外观"栏目下,看看有什么变化,是不是多了个预览图呢?主题制作完毕后,你可以给你的主题界面截个图,命名为 screenshot.png,大小300 * 225,放置到主题目录下作为主题缩略图。

好了,今天我们要讲的就这些,敬请期待下文...

最后将 [WordPress 主题制作全过程\(三\):HTML 静态模板制作](#) 下载到的 Aurelius 目录下的所有文件拷贝到 wp-content\themes\Aurelius 目录下,并将所有 .html 后缀改成 .php,以备我们下节内容使用。

WordPress 主题制作全过程(五):制作 header.php

你可以尝试用文本编辑器打开从 [WordPress 主题制作全过程\(三\):HTML 静态模板制作](#) 下载到的 .html 文件,不知道你有没有发现他们头部的代码都非常的相似呢?其实我们可以提取这部分相似的代码,放到一个单独的文件 header.php 中,各个页面想用这部分代码的时候再用 php 的 include 包含进去,省的每个页面里面都要写这部分代码,更改起来也可以达到一改全改的目的。

再次提醒:如果你不打算动手编写代码,这个系列教程就别看了,对你无益!

接着我们上次创建的主题目录 wp-content\themes\Aurelius,在该目录下新建一个 php 文件 header.php,我们提取出 index.php 中的头部代码复制粘贴到 header.php 中,下面是的代码就是目前 header.php 中的所有代码了(当然不同主题的头部代码都是不一样,在你实际的项目中可以自定决定):

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head profile="http://gmpg.org/xfn/11">
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 <title>Aurelius | Blog</title>
7 <!-- Stylesheets -->
8 <link rel="stylesheet" href="/style.css" type="text/css" media="screen" />
9 </head>
10 <body>
11 <div id="wrapper" class="container_12 clearfix">
12     <!-- Text Logo -->
13     <h1 id="logo" class="grid_4">Aurelius</h1>
14     <!-- Navigation Menu -->
15     <ul id="navigation" class="grid_8">
16         <li><a href="contact.html"><span class="meta">Get in touch</span> <br />
17             Contact Us</a></li>
18         <li><a href="blog.html" class="current"><span class="meta">Latest
19 news</span> <br />
20             Blog</a></li>
21         <li><a href="index.html"><span class="meta">Homepage</span> <br />
22             Home</a></li>
```

```
23     </ul>
24     <div class="hr grid_12 clearfix">&nbsp;</div>
25     <!-- Caption Line -->
    <h2 class="grid_12 caption clearfix">Our <span>blog</span>, keeping you
up-to-date on our latest news.</h2>
    <div class="hr grid_12 clearfix">&nbsp;</div>
```

再用文本编辑器打开 index.php、archive.php、contact.php、full_width.php、page.php 和 single.php，删掉以上类似代码，改成：

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

好，现在打开你的测试博客主页，看看我们制作的主题是否还可以正常工作，答案是可以的，跟原来几乎没什么两样，但还是一片混乱。get_header()就相当于将 header.php 中的代码拷贝到当前的 php 文件。接下来，我们将仔细探讨 header.php 中的动态内容。header.php 将会被所有的模板页面（主页、分类页、页面、标签页等）所包含，所以 header.php 中代码应该是动态，适合不同页面的，所以这里面需要用到 PHP 代码，而不是单纯的 HTML。下面让我们一起来修改 header.php：

1、更改<title>

我们都知道不同页面的 title 都是不一样，而且 title 的设置还会直接影响到 SEO 的效果，所以这里应该谨慎设置。下面提供一种 SEO 优化的 title 写法，将<title>Aurelius | Blog</title>改成：

```
<title><?php if ( is_home() ) {  
1      bloginfo('name'); echo " - ";  
2  bloginfo('description');  
3      } elseif ( is_category() ) {  
4          single_cat_title(); echo " - ";  
5  bloginfo('name');  
6      } elseif (is_single() || is_page() ) {  
7          single_post_title();  
8      } elseif (is_search() ) {  
9          echo "搜索结果"; echo " - ";  
10  bloginfo('name');  
11      } elseif (is_404() ) {  
12          echo '页面未找到!';  
13      } else {  
          wp_title('','true');  
      } ?></title>
```

以上添加的 php 代码运用了条件判断，针对不同的页面采用不同 title，这里解释一下这几个条件标签。

- is_home() ：当前页面为主页时返回 true
- is_category() ：当前页面为分类页时返回 true
- is_single() ：当前页面为单文章页时返回 true
- is_page() ：当前页面为单页面时返回 true

- 更详细的内容参阅 WordPress 文档：[条件标签](#)

到目前为止，可能你对这些条件判断标签还没有深入的认识，也搞不懂到底是用了这些标签会对主题造成怎样的影响的，随着我们教程的进一步深入，你会慢慢理解的。

2、更改样式表 style.css 路径

在此之前你看到的首页都是混乱的，原因是还没加载 css 样式。现在我们一起把样式加上。你可以在 header.php 中找到这一段代码：

```
1 <link rel="stylesheet" href="./style.css" type="text/css"
  media="screen" />
```

聪明的你可能问：wp-content\themes\Aurelius 目录下不是已经有一个 style.css 吗？那为什么 header.php 没有加载 css 呢？结果你是可以看到的，页面一片混乱，可以确定确实没有加载到 css。因为这是 WordPress 的主题，是要被 WordPress 的主程序调用，经过层层解析才能把你的博客显示出来，而不是简简单单的 html 静态网页文件。正确的改法：

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

bloginfo('stylesheet_url') 输出的是你的主题 css 文件绝对网址，如 <http://localhost/wp/wp-content/themes/Aurelius/style.css>，WordPress 程序会自动识别你的 WordPress 安装地址，当前启用的主题，自动输出这个 style.css 链接。现在你可以试着更改一下，然后刷新一下你的博客首页，查看网页源代码，style.css 的链接是不是变成你的了？页面是否可以正常显示了呢？

不过，还有几张图片的路径不对，还不能显示出来，现在我们一起用文本编辑器打开 index.php、archive.php、contact.php、full_width.php、page.php 和 single.php，给这些图片加上正确的 URL，搜索代码，将所有的：`src="images/`，批量替换成 `src="<?php bloginfo('template_url'); ?>/images/`。现在再刷新你的主页，看文章的缩略图610×150是否可以正常显示。`<?php bloginfo('template_url'); ?>`用于输出主题目录的 URL。

3、添加 pingback

至于什么是 pingback，你可以在搜索引擎中输入关键字 WordPress pingback，就可以得到你想要的答案了。如果你需要这个功能，可以将在<head>里面添加代码：

```
1 <link rel="pingback" href="<?php  
    bloginfo('pingback_url'); ?>" />
```

4、更改博客名称和描述

在 header.php，下面两行代码用于显示博客名称和描述：

```
1 <h1 id="logo" class="grid_4">Aurelius</h1>  
2 <h2 class="grid_12 caption clearfix">Our <span>blog</span>, keeping you up-to-date  
    on our latest news.</h2>
```

上面是静态代码，现在做如下修改：

```
1 <h1 id="logo" class="grid_4"><a href="<?php echo get_option('home'); ?>/"><?php  
2 bloginfo('name'); ?> </a> </h1>  
  
    <h2 class="grid_12 caption clearfix"><?php bloginfo('description'); ?> </h2>
```


现在你的博客首页看到的就是你博客名称和描述了，并且 logo 也是一个链接指向你的博客首页。我们这里说说这些 php 代码的作用。

- `<?php echo get_option('home'); ?>` 输出你的博客首页网址
- `<?php bloginfo('name'); ?>` 输出你的博客名称
- `<?php bloginfo('description'); ?>` 输出博客描述

博客名称和描述可以在 WordPress 管理后台 - 设置 - 常规那里更改。以后制作你自己的 WordPress 主题的时候，你可参照上面的说明对你的主题进行修改。

5、添加订阅 feed 链接

相信每个已发布的 WordPress 博客主题都会提供 feed 订阅，当然我们的主题也应该提供这样的功能。在 `</head>` 之前添加以下代码：

```
1 <link rel="alternate" type="application/rss+xml" title="RSS 2.0 - 所有文章" href="<?php  
echo get_bloginfo('rss2_url'); ?>" />  
2 <link rel="alternate" type="application/rss+xml" title="RSS 2.0 - 所有评论" href="<?php  
bloginfo('comments_rss2_url'); ?>" />
```

6、添加 wp_head

有些插件需要在网页头部添加一些 js 或 css，要让这些插件能够正常的工作，也让你的主题有更好的兼容性，你应该添加 `wp_head()` 函数。打开 `header.php`，在 `</head>` 前面添加以下代码即可：

```
1 <?php
```

```
wp_head(); ?>
```

现在打开你的博客主页，查看源代码，</head>前面是不是多了以下类似代码（这些都是 wp_head()的功劳）：

```
1 <link rel="EditURI" type="application/rsd+xml" title="RSD"
2 href="http://ludou.co.tv/blog/xmlrpc.php?rsd" />
3 <link rel="wlwmanifest" type="application/wlwmanifest+xml"
4 href="http://ludou.co.tv/blog/wp-includes/wlwmanifest.xml" />
5 <link rel='index' title='露兜实验室' href='http://ludou.co.tv' />
6 <meta name="generator" content="WordPress 2.9.2" />
```

7、添加 Description 和 Keywords

关于添加网页描述和关键字，可以查看我之前写过的文章：[WordPress 使用经验\(一\)](#)

[独立的 Description 和 Keywords](#)

8、显示菜单栏

目前菜单栏有 Home、Blog 和 Contact Us 几个菜单，不过这些都是静态的内容，并不是你博客上的页面。现在我们将菜单栏换成你的菜单，这里只在菜单栏中列出页面 page，当然你也可以再放置分类，根据你的喜好来吧，将 header.php 中：

```
1 <ul id="navigation" class="grid_8">
2     <li><a href="contact.html"><span class="meta">Get in
3 touch</span><br />
```

```
4         Contact Us</a></li>
5
6     <li><a href="blog.html" class="current"><span class="meta">Latest
7 news</span><br />
8         Blog</a></li>
9
10    <li><a href="index.html"><span class="meta">Homepage</span><br
11 />
12
13        Home</a></li>
14
15</ul>
```

改成：

```
1 <ul id="navigation" class="grid_8">
2
3     <?php wp_list_pages('depth=1&title_li=0&sort_column=menu_order'); ?>
4
5     <li <?php if (is_home()) { echo 'class="current"'; } ?>><a title="<?php
6 bloginfo('name'); ?>" href="<?php echo get_option('home'); ?>/">主页</a></li>
7
8
9</ul>
```

具体如何在菜单栏显示分类,你可以看这篇文章,有什么问题再给我留言 :[WordPress 分类做导航栏,并高亮显示](#)

另外,可以参考这两个函数的说明(英文):

- [Template Tags/wp list pages](#)
- [Template Tags/wp list categories](#)

9、刷新缓存

在<body>前面 ,</head>后面添加 PHP 代码 ,用于提高程序运行效率 :<?php flush(); ?>

总结

好了 , 本次练习到此结束 ! 现在总结一些今天讲到的比较重要的知识点 :

- **<?php get_header(); ?>** 从当前主题文件夹中包含 header.php 文件
- **is_home() , is_single() , is_category()**等几个条件判断标签
- **<?php bloginfo('stylesheet_url'); ?>** 输出主题文件夹中 style.css 文件的路径
- **<?php bloginfo('pingback_url'); ?>** 输出博客 pingback 网址
- **<?php bloginfo('template_url'); ?>** 输出博客主题目录 URL
- **<?php echo get_option('home'); ?>** 输出你的博客首页网址
- **<?php bloginfo('name'); ?>** 输出你的博客名称
- **<?php bloginfo('description'); ?>** 输出博客描述
- **<?php wp_head(); ?>** 用于包含 WordPress 程序输出头部信息
- **<?php wp_list_categories(); ?>** 用于列出博客分类页
- **<?php wp_list_pages(); ?>** 用于列出博客页面

到目前为止你的博客还只能看到主页 , 不要灰心 , 凡事一步一个脚印 , 以后教程会慢慢深入的。最后提供经过本次修改后的 Aurelius 主题文件 , 你可以用文本编辑器打开看看 , 跟你修改的文件比较比较 (尤其是 header.php) , 看看你改得怎么样 ?

下载

这个文件

WordPress 主题制作全过程 (六) : 制作 footer.php

上节内容我们已经弄好了博客的页头 header.php , 今天我们就一起来制作页脚

footer.php。footer.php 与 header.php 差不多，写这个文件的目的是为了精简代码，提高代码的重用性，Aurelius 主题目录中的所有页面的页脚代码几乎都是一样的，我们就把这些代码提取出来放到 footer.php。

在 Aurelius 主题目录中新建文件 footer.php，我们提取出 index.php 中的页脚代码拷贝到 footer.php 中(如果以后你制作自己的主题，可以根据需要决定哪些是 footer 代码)：

```
<!-- Footer -->  
1 <p class="grid_12 footer clearfix"> <span class="float"><strong>Design  
2 By</strong> QwibbleDesigns&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<strong>Code  
3 By</strong> <a href="http://www.ludou.org/">Ludou</a></span> <a class="float  
4 right" href="#">top</a> </p>  
5 </div>  
6 <!--end wrapper-->  
  
</body>  
  
</html>
```

再用文本编辑器打开 index.php、archive.php、contact.php、full_width.php、page.php 和 single.php，删掉以上类似代码，改成：

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

好，现在打开你的博客主页，看看我们制作的主题是否还可以正常工作，答案是可以的，跟原来几乎没什么两样，页脚还是跟原来一样。现在来添加上你的博客版权声明信息和 wp_footer()函数，将 footer.php 中所有代码删除，改成：

这是对他们最起码的尊重！

好了，footer.php 的制作就这么简单。

最后提供经过本次修改后的 Aurelius 主题文件，你可以用文本编辑器打开看看，跟你修改的文件比较比较，看看你改得怎么样？



WordPress 主题制作全过程（七）：制作 sidebar.php

制作好了 header.php 和 footer.php，今天我们来制作侧边栏 sidebar.php。由于侧边栏的可定制性实在是太强了，所以本节内容比较难，我讲解起来也比较困难，有些内容会被略掉！

作为各个页面公用的侧边栏，我们还是像制作 header.php 和 footer.php 那样，从 index.php 中提取侧边栏，放到 sidebar.php。好，现在在你的主题目录 Aurelius 下新建文件 sidebar.php，从 index.php 中提取一下代码，放到 sidebar.php 中：

```
1      <!-- Column 2 / Sidebar -->
2
3      <div class="grid_4">
4
5          <h4>Catagories</h4>
6
7          <ul class="sidebar">
8              <li><a href="">So who are
we?</a></li>
              <li><a
href="">Philosophy</a></li>
```

```
9         <li><a
10 href="">History</a></li>
11         <li><a
12 href="">Jobs</a></li>
13         <li><a
14 href="">Staff</a></li>
15         <li><a
16 href="">Clients</a></li>
17     </ul>
18     <h4>Archives</h4>
19     <ul class="sidebar">
20         <li><a href="">January
21 2010</a></li>
22         <li><a href="">December
23 2009</a></li>
24         <li><a href="">Novemeber
25 2009</a></li>
26         <li><a href="">October
27 2009</a></li>
28         <li><a href="">September
29 2009</a></li>
30         <li><a href="">August
```



```
2009</a></li>

</ul>

</div>

<div class="hr grid_12
clearfix">&nbsp;</div>
```

再用文本编辑器打开 index.php、archive.php、page.php 和 single.php，删掉以上类似代码，改成：

```
1 <?php
   get_sidebar(); ?>
```

好，现在打开你的博客主页，看看我们制作的主题是否还可以正常工作。现在我们的侧边栏还都是静态的代码，大家可能都知道在 WordPress 后台 – 外观 – 小工具，那里可以拖动你想要的栏目到侧边栏，但是我们的主题目前还不支持这个功能。现在就让我一起来制作完整的 sidebar。

为了适应 WordPress 程序，我们还要对 sidebar.php 做一些微调，下载新的样式表 style.css，替换 Aurelius 目录下的 style.css，[点此下载 style.css](#)

开始 sidebar.php 的制作，我们将在侧边栏放置4个栏目。在初始状态下，也就是你没有在侧边栏放置任何小工具的情况下，这4个栏目自上而下为分类目录、最新文章、标签云和文章月存档。现在将 sidebar.php 中所有代码删除，改成：

```
1 <!-- Column 2 / Sidebar -->
2 <div class="grid_4">
```

```
3
4     <?php if ( !function_exists('dynamic_sidebar')
5         || !dynamic_sidebar('First_sidebar') ) : ?>
6         <h4>分类目录</h4>
7         <ul>
8             <?php
9 wp_list_categories('depth=1&title_li=&orderby=id&show_count=0&hide_empty=1&chi
10 ld_of=0'); ?>
11         </ul>
12     <?php endif; ?>
13
14     <?php if ( !function_exists('dynamic_sidebar')
15         || !dynamic_sidebar('Second_sidebar') ) : ?>
16         <h4>最新文章</h4>
17         <ul>
18             <?php
19                 $posts = get_posts('numberposts=6&orderby=post_date');
20                 foreach($posts as $post) {
21                     setup_postdata($post);
22                     echo '<li><a href="' . get_permalink() . '">' . get_the_title() .
23 '</a></li>';
24                 }
```

```
25         $post = $posts[0];
26     ?>
27 </ul>
28 <?php endif; ?>
29
30 <?php if ( !function_exists('dynamic_sidebar')
31         || !dynamic_sidebar("Third_sidebar") ) : ?>
32     <h4>标签云</h4>
33     <p><?php wp_tag_cloud('smallest=8&largest=22'); ?> </p>
34 <?php endif; ?>
35
36 <?php if ( !function_exists('dynamic_sidebar')
37         || !dynamic_sidebar("Fourth_sidebar") ) : ?>
38     <h4>文章存档</h4>
39     <ul>
40         <?php wp_get_archives('limit=10'); ?>
41     </ul>
42 <?php endif; ?>
</div>
<div class="hr grid_12 clearfix">&nbsp;</div>
```

然后[点此处下载 functions.php](#) 放到主题目录 Aurelius 下，这时候你的侧边栏就可

以正常工作了,在 WordPress 后台 – 外观 – 小工具,可以正常地拖动小工具到侧边栏了。

好了, sidebar.php 到此就制作成功了。这节内容理论的东西讲得不多,关键是我不知道如何去表述这些东西,就算我说出来,可能你也不会太明白了。关于 sidebar.php 制作的技术文档:可以参考:

- [Customizing Your Sidebar](#)
- [Function Reference/register sidebar](#)

最后还是按照惯例,提供经过本次修改后的主题文件下:



WordPress 主题制作全过程 (八): 制作 index.php

前面我们已经制作完成博客所有公用页面 sidebar.php、footer.php 和 header.php,今天开始我们就要制作单独的页面了。现在我们要制作的是索引页 index.php,这里我们可以暂时把它理解成主页,但事实上它并不是首页这么简单([点此查看原因](#))。

在主页中主要就是文章列表,将你博客上的文章一篇一篇地列出来。你可能已经注意到,主页中每篇文章的样式都是一样的,只是标题、时间、作者和摘要等文字内容不一样而已,嗯!我们制作 index.php 也只需要一篇文章的 html 代码,不需要手动地去写那么多文章的 html,况且这样也不是动态的内容。我们只需要一个循环就可以将所有文章显示到首页上,循环就是重复做某件事情,这里的循环是重复地输出文章。如果你之前学过任一门计算机程序设计语言的话,就不难理解什么是循环,循环的作用也一想就通,如 while, for, foreach.....

在这里插一句,如果你真的想了解如何制作主题,请打开的文本编辑器,跟着我一步

一步地操作，一步一步地修改，每做一次修改就刷新一下你的博客看看有什么变化，这样才能加深你的理解。如果你懒得动手，建议以后的内容就不用看了，看了对你帮助也不大。

现在开始制作 index.php。初始情况下 index.php 中有三篇文章，打开 index.php 你可以看到文章的3个标记<!-- Blog Post -->，我们现在其他将两篇文章的代码删除，留下一篇，并将文章摘要去除。修改后的代码是这样的：

```
1 <?php get_header(); ?>
2 <!-- Column 1 /Content -->
3 <div class="grid_8">
4 <!-- Blog Post -->
5 <div class="post">
6 <!-- Post Title -->
7 <h3 class="title"><a href="single.html">Lorem ipsum massa cras
8 phasellus</a></h3>
9 <!-- Post Data -->
10 <p class="sub"><a href="#">News</a>, <a href="#">Products</a> &bull;
11 31st Sep, 09 &bull; <a href="#">1 Comment</a></p>
12 <div class="hr dotted clearfix">&nbsp;</div>
13 <!-- Post Image -->
14 
16 <!-- Post Content -->
```

```
17
18         <!-- Read More Button -->
19         <p class="clearfix"><a href="single.html" class="button right"> Read
20 More...</a></p>
21     </div>
22     <div class="hr clearfix">&nbsp;</div>
23
24     <!-- Blog Navigation -->
25
26     <p class="clearfix"> <a href="#" class="button float">&lt;&lt; Previous
27 Posts</a> <a href="#" class="button float right">Newer Posts >></a> </p>
28
29     </div>
30
31     <?php get_sidebar(); ?>
32
33 <?php get_footer(); ?>
```

从上面的代码可以看出，一篇文章的 html 骨架就是：

```
1 <div class="post">
2     <!-- Post Title -->
3     <h3 class="title"><a href="single.html">文章标题</a></h3>
4     <!-- Post Data -->
5     <p class="sub"><a href="#">标签1</a>, <a href="#">标签12</a> &bull; 发布时间
6 &bull; <a href="#">评论数</a></p>
7     <div class="hr dotted clearfix">&nbsp;</div>
```

```
8      <!-- Post Image 文章的缩略图 -->
9
10     
12
13     <!-- Post Content -->
14     文章内容
15
16     <!-- Read More Button -->
17
18     <p class="clearfix"><a href="single.html" class="button right"> 阅读全文按钮
19
20     </a></p>
21
22     </div>
23
24     <div class="hr clearfix">&nbsp;</div>
```

不同主题的主题的文章 html 骨架是不一样的，如果你熟悉 html，可以很快地分辨出文章骨架，以上是我们这个主题的骨架，我们将以此为基础给 index.php 加上动态内容：

1、添加文章标题

找到：

```
1 <h3 class="title"><a href="single.html">Lorem ipsum massa cras
   phasellus</a></h3>
```

改成：

```
1 <h3 class="title"><a href="<?php the_permalink(); ?>" rel="bookmark"><?php
   the_title(); ?></a></h3>
```

这里解释一下这几个 php 函数：

- 输出文章的 URL 链接 ([参考文档](#))
- 输出文章的标题 ([参考文档](#))

2、添加文章标签

我们很多人在写文章的时候都喜欢添加一些标签 ,况且侧边栏我们也加入了"标签云",

我们的主题应该支持标签。找到 :

```
1 <a href="#">News</a>, <a  
href="#">Products</a>
```

改成 :

```
1 <?php the_tags("标签: ',' ,  
"); ?>
```

函数参考 : [the_tags](#)

3、添加日期

找到 : 31st Sep, 09

改成 :

```
1 <?php the_time('Y 年 n 月 j  
日') ?>
```

函数参考 : [the_time](#)

关于该函数中 Y n j 获取的日期格式 , 你可以参考文档 (中文) , 选择你喜欢的时间

格式：[zh-cn:自定义时间和日期](#)

可能你看了以上提供的时间和日期文档，还是一头雾水，下面提供几个示例，你就差不多能够依样画葫芦，指定自己喜欢的时间日期格式：

PHP 代码	输出内容
<code><?php the_time('Y 年 n 月 j 日') ?></code>	1999年5月1日
<code><?php the_time('Y 年 m 月 d 日') ?></code>	1999年05月01日
<code><?php the_time('Y-m-d') ?></code>	1999-05-01
<code><?php the_time('y-m-d H:i:s') ?></code>	99-05-01 02:09:08

4、显示评论数

现在我们来添加评论数代码，查找代码：

```
1 <a href="#">1  
   Comment</a>
```

改成：

```
1 <?php comments_popup_link('0 条评论', '1 条评论', '% 条评论', '', '评论已关  
   闭'); ?>
```

该函数会根据文章的评论数量显示不同的文字链接，0 条评论、1 条评论等等，当然你可以根据自己的爱好定制文字内容。该链接会直接打开对应的文章，并移动到文章的评论区，你可以点此查看效果：[2 条评论](#)

函数参考：[comments_popup_link](#)

5、添加编辑按钮

如果文章作者已登录，我们将允许他在首页点击对应文章的编辑按钮，就可以直接修改文章，这个功能是可选的，你可以不添加。接上面的评论按钮，我们在其后面添加相应代码：

```
1 <?php comments_popup_link('0 条评论', '1 条评论', '% 条评论', '', '评论已关闭'); ?><?php  
edit_post_link('编辑', ' &bull; ', ''); ?>
```

函数参考：[edit_post_link](#)

6、添加文章内容

可能有些人喜欢在首页输出全文，而有些人喜欢在首页输出文章摘要，这里提供两种方案，任君选择。查找：<!-- Post Content -->

将其改成：

```
1 <!-- Post Content -->  
2 <?php  
the_excerpt(); ?>
```

只要在写文章的时候在"摘要"框内填写摘要，在首页显示的就是摘要，如果不填就输出全文！以下是方案二，用于输出全文，除非你在文章中使用了<!-- more -->，代码修改：

```
1 <!-- Post Content -->  
2 <?php the_content('阅读全文...'); ?>
```

函数参考:

- [the_excerpt](#)
- [the_content](#)

7、阅读全文

这里给添加阅读全文链接，如果在**6、添加文章**内容中你选择了输出全文，本步骤可以忽略，查找代码：

```
1 <a href="single.html" class="button right"> Read  
More...</a>
```

改成：

```
1 <a href="<?php the_permalink(); ?>" class="button right"> 阅读全文  
</a>
```

8、添加文章循环

到目前为止，你的首页还只能显示一篇文章，要想输出所有文章，需要我们之前提到的循环。查找代码：

```
<!-- Blog Post -->
```

改成：

```
1 <!-- Blog Post -->  
2 <?php if (have_posts()) : while (have_posts()) :
```

```
the_post(); ?>
```

再查找：

```
1 <div class="hr  
clearfix">&nbsp;</div>
```

改成：

```
1 <div class="hr  
2 clearfix">&nbsp;</div>  
  
    <?php endwhile; ?>
```

再次查找：

```
1 </div>  
2 <?php  
get_sidebar(); ?>
```

改成：

```
1 <?php else : ?>  
2 <h3 class="title"><a href="#" rel="bookmark">未找到  
3 </a> </h3>  
4 <p>没有找到任何文章！ </p>  
5 <?php endif; ?>  
6 </div>  
  
    <?php get_sidebar(); ?>
```

好了，现在查看你的主页，是不是可以显示多篇文章了呢？文章数量取决于你在后台设置每页可显示的文章数量。以上的循环可以简化为以下内容，这样看起来应该比较容易理解了，在 endwhile 之前不断地输出每篇文章，直至文章数量达到每页显示的最大文章数量；如果你的博客上一篇文章都没有，就会输入无文章提示。

```
1 <?php if (have_posts()) : while (have_posts()) :  
2 the_post(); ?>  
3     文章 html 骨架  
4 <?php endwhile; ?>  
5 <?php else : ?>  
6     输出找不到文章提示  
    <?php endif; ?>
```

参考文档: [The Loop](#)

9、添加文章分页

上面你已经看到，每页只能显示部分文章，要想看下一页，就得添加分页。现在查找代码：

```
1 <p class="clearfix"> <a href="#" class="button float">&lt;&lt; Previous Posts</a> <a  
    href="#" class="button float right">Newer Posts >></a> </p>
```

改成：

```
1 <p class="clearfix"> <?php previous_posts_link('&lt;&lt; 查看新文章', 0); ?> <span  
    class="float right"> <?php next_posts_link('查看旧文章 &gt;&gt;', 0); ?> </span> </p>
```

参考函数：

- [previous_posts_link](#)
- [next_posts_link](#)

10、文章缩略图

对于大部分人来说，不太需要文章缩略图的功能，而且有很多插件可以实现这个功能。

这里我们将首页的文章缩略图代码删除：

```
1 <!-- Post Image -->
2
    
```

另外，还有个存档页面的模板 archive.php，跟 index.php 的制作过程完全一样，只不过需要在 functions.php 里添加一个函数，这里就不再罗嗦，下载自己看吧，注意：functions.php 中的代码已经修改，要想让你的分类、标签等存档页能够正常显示，请下载最新的 functions.php 覆盖原来的。另外，标签页和分类页支持在该页面顶部显示介绍，前提是你后台文章标签和分类处要填上了**描述**。

好了，到目前这个主题也像个样子了，不过还有很多要完善，后面我们将继续完善！

按照惯例，继续提供经过本次修改后的主题文件下载：



WordPress 主题制作全过程 (九)：制作 single.php

Ludou 评分 9.9, 满分 10 分(21 票) **评分**

人气: 8,403 [添加评论\(29 \)](#)

今天我们来制作单文章页 single.php，有了之前制作 index.php 的经验，制作 single.php 也不再那么难了，这里将直接略过一些内容，直接给出结果。如果对某些修改不太清楚，可以先参考：[WordPress 主题制作全过程（八）：制作 index.php](#)

1、添加文章标题：

```
1 <h3 class="title"><a href="single.html">Lorem ipsum massa cras  
phasellus</a></h3>
```

改成：

```
1 <h3 class="title"><a href="<?php the_permalink(); ?>"><?php  
the_title(); ?></a></h3>
```

2、添加文章标签

```
1 <a href="#">News</a>, <a  
href="#">Products</a>
```

改成：

```
1 <?php the_tags('标签: ', ' ',  
"); ?>
```

3、添加日期

找到：31st Sep, 09 改成：

```
1 <?php the_time('Y 年 n 月 j  
日') ?>
```

4、显示评论数

```
1 <a href="#">7  
Comments</a>
```

改成：

```
1 <?php comments_popup_link('0 条评论', '1 条评论', '% 条评论', '', '评论已关  
闭'); ?>
```

5、添加编辑按钮

接上面的评论代码，改成：

```
1 <?php comments_popup_link('0 条评论', '1 条评论', '% 条评论', '', '评论已关闭'); ?><?php  
edit_post_link('编辑', ' &bull; ', ''); ?>
```

6、添加文章内容

将<!-- Post Content --> 和 <!-- Post Links -->之间的代码全部删除，替换成：

```
1 <?php
```



```
the_content(); ?>
```

另外，你可以将文章页那张图片删除了，删除以下代码：

```
1 
```

7、添加返回博客首页和发表评论按钮

其实就是添加博客首页和评论锚点链接，在制作 header.php，我们已经知道可以通过 `get_option('home');` 来获取博客地址。

```
1 <p class="clearfix"> <a href="blog.html" class="button float"> &lt;&lt; Back to  
Blog </a> <a href="#commentform" class="button float right"> Discuss this post </a>  
</p>
```

改成：

```
1 <p class="clearfix"> <a href="<?php echo get_option('home'); ?>" class="button float"  
> &lt;&lt; 返回首页 </a> <a href="#commentform" class="button float right"> 发表评论  
</a> </p>
```

好了，基本上的修改就这些了，但是你的文章页仍然不能显示文章内容，你得给它加上一个条件语句，这样 WordPress 才会去数据库读出你的文章内容。搜索代码： `<!-- Column 1 /Content -->`

改成：

```
1 <!-- Column 1 /Content -->
2     <?php if (have_posts()) : the_post();
    update_post_caches($posts); ?>
```

将：

```
1 </div>
2     <?php
    get_sidebar(); ?>
```

改成：

```
1     </div>
2     <?php else : ?>
3     <div
4     class="errorbox">
5         没有文章！
6     </div>
7     <?php endif; ?>
    <?php
    get_sidebar(); ?>
```

现在你的文章内容应该都可以正常显示了，一个文章页基本上也成型了。下节我们将讲解如何制作评论页，本次不提供修改的主题文件下载，下次一起提供。

另外，文章页顶部会有一段文字：

Our blog, keeping you up-to-date on our latest news.

可以替换成你的内容。如果不需要，可以将以下代码删除：

```
1 <h2 class="grid_12 caption clearfix">Our <span>blog</span>, keeping you up-to-date
2 on our latest news.</h2>

<div class="hr grid_12 clearfix">&nbsp;</div>
```

WordPress 主题制作全过程 (十): 制作 comments.php

今天我们来制作评论主题的评论模块。在主题目录 Aurelius 下新建 comments.php 在 single.php 剪切以下代码，粘贴到 comments.php：

```
1 <!-- Comment's List -->
2 <h3>Comments</h3>
3 <div class="hr dotted clearfix">&nbsp;</div>
4 <ol class="commentlist">
5     <li class="comment">
6         <div class="gravatar"> <img alt="" src='images/gravatar.png' height='48'
7 width='48' /> <a class="comment-reply-link" href=&quot;>Reply</a> </div>
8         <div class="comment_content">
9             <div class="clearfix"> <cite class="author_name"><a href="">Joe
10 Bloggs</a> </cite>
11             <div class="comment-meta commentmetadata">January 6, 2010
12 at 6:26 am</div>
13         </div>
14         <div class="comment_text">
```

```
15         <p>Donec leo. Aliquam risus elit, luctus vel, interdum vitae,
16 malesuada eget, elit. Nulla vitae ipsum. Donec ligula ante, bibendum sit amet,
17 elementum quis, viverra eu, ante. Fusce tincidunt. Mauris pellentesque, arcu eget feugiat
18 accumsan, ipsum mi molestie orci, ut pulvinar sapien lorem nec dui.</p>
19     </div>
20 </div>
21 </li>
22 </ol>
23 <div class="hr clearfix">&nbsp;</div>
24 <!-- Comment Form -->
25 <form id="comment_form" action="" method="post">
26     <h3>Add a comment</h3>
27     <div class="hr dotted clearfix">&nbsp;</div>
28     <ul>
29         <li class="clearfix">
30             <label for="name">Your Name</label>
31             <input id="name" name="name" type="text" />
32         </li>
33         <li class="clearfix">
34             <label for="email">Your Email</label>
35             <input id="email" name="email" type="text" />
36         </li>
```

```
37         <li class="clearfix">
38             <label for="email">Your Website</label>
39             <input id="website" name="website" type="text" />
40         </li>
41         <li class="clearfix">
42             <label for="message">Comment</label>
43             <textarea id="message" name="message" rows="3"
cols="40"></textarea>
44         </li>
45         <li class="clearfix">
46             <!-- Add Comment Button -->
47             <a type="submit" class="button medium black right">Add comment</a>
48         </li>
49     </ul>
50 </form>
```

在 single.php 原位置添加代码：

```
1 <?php
comments_template(); ?>
```

以上语句的作用就是将 comments.php 里的所有内容导入到 single.php 中，与直接在 single.php 写 comments.php 中的代码效果是一样的。

为了安全起见，不让恶意用户直接打开评论文件，请在 comments.php 头部添加以

下代码：

```
1 <?php
2     if (isset($_SERVER['SCRIPT_FILENAME']) && 'comments.php' ==
3         basename($_SERVER['SCRIPT_FILENAME']))
4         die ('Please do not load this page directly. Thanks!');
5
6 ?>
```

因为 WordPress 的输出评论函数 wp_list_comments() 输出的评论代码与我们主题的评论代码不一样的，我们得自定义我们的评论列表，将 comments.php 中的以下代码删除（以下代码用于列出文章的所有评论）：

```
1 <li class="comment">
2     <div class="gravatar"> <img alt="" src='images/gravatar.png' height='48'
3         width='48' /> <a class="comment-reply-link" href="#">Reply</a> </div>
4     <div class="comment_content">
5         <div class="clearfix"> <cite class="author_name"> <a href="#">Joe
6         Bloggs</a> </cite>
7         <div class="comment-meta commentmetadata">January 6, 2010 at 6:26
8         am</div>
9     </div>
10    <div class="comment_text">
11        <p>Donec leo. Aliquam risus elit, luctus vel, interdum vitae, malesuada eget, elit.
12        Nulla vitae ipsum. Donec ligula ante, bibendum sit amet, elementum quis, viverra eu,
```

```
ante. Fusce tincidunt. Mauris pellentesque, arcu eget feugiat accumsan, ipsum mi  
molestie orci, ut pulvinar sapien lorem nec dui.</p>  
  
</div>  
  
</div>  
  
</li>
```

改成：

```
1 <?php  
2 if (!empty($post->post_password) && $_COOKIE['wp-postpass_'. COOKIEHASH] !=  
3 $post->post_password) {  
4     // if there's a password  
5     // and it doesn't match the cookie  
6     ?>  
7     <li class="decmt-box">  
8         <p><a href="#addcomment">请输入密码再查看评论内容.</a></p>  
9     </li>  
10    <?php  
11        } else if ( !comments_open() ) {  
12        ?>  
13        <li class="decmt-box">  
14            <p><a href="#addcomment">评论功能已经关闭!</a></p>  
15        </li>
```

```
16     <?php
17         } else if ( !have_comments() ) {
18     ?>
19     <li class="decmt-box">
20         <p><a href="#addcomment">还没有任何评论，你来说两句吧</a></p>
21     </li>
22     <?php
23         } else {
24             wp_list_comments('type=comment&callback=aurelius_comment');
25         }
26     ?>
```

以上代码的意思大致也可以看得出来了，就是一大堆 **如果...就....**，如果以上条件都不满足就列出所有评论。现在将主题文件夹 Aurelius 中的 functions.php 中的 ?> ，改成以下代码，如果你之前从本博客下载到的 functions.php 已经有以下代码则不用再添加：

```
1 function aurelius_comment($comment, $args, $depth)
2 {
3     $GLOBALS['comment'] = $comment; ?>
4     <li class="comment" id="li-comment-<?php comment_ID(); ?>">
5         <div class="gravatar"> <?php if (function_exists('get_avatar') &&
6 get_option('show_avatars')) { echo get_avatar($comment, 48); } ?>
7     <?php comment_reply_link(array_merge( $args, array('reply_text' => '回复','depth' =>
```



```
8 $depth, 'max_depth' => $args['max_depth']))) ?> </div>
9
10 <div class="comment_content" id="comment-<?php comment_ID(); ?>">
11
12 <div class="clearfix">
13
14 <?php printf(__('<cite class="author_name">%s</cite>'),
15 get_comment_author_link()); ?>
16
17 <div class="comment-meta commentmetadata">发表于: <?php
18 echo get_comment_time('Y-m-d H:i'); ?> </div>
19
20 &nbsp;&nbsp;&nbsp;&nbsp;<?php edit_comment_link('修改'); ?>
21
22 </div>
23
24 <div class="comment_text">
25
26 <?php if ($comment->comment_approved == '0') : ?>
27
28 <em>你的评论正在审核，稍后会显示出来！ </em> <br />
29
30 <?php endif; ?>
31
32 <?php comment_text(); ?>
33
34 </div>
35
36 </div>
37
38 </li>
39
40 <?php } ?>
```

以上代码所用到的 WordPress 函数及相应的说明：

函数名称	函数功能
------	------

<code>get_avatar(\$comment, 48)</code>	获取评论者的 gravatar 头像 , 尺寸为48 * 48
<code>comment_reply_link()</code>	回复留言的链接
<code>get_comment_author_link</code>	用于获取评论者博客地址
<code>get_comment_time</code>	获取评论发布时间
<code>edit_comment_link</code>	管理员修改评论的链接
<code>comment_text()</code>	输出评论内容

好, 现在在你的文章页面底部就可以正常地显示评论了! 现在我们继续来制作提交评论的表单, 将以下代码删除 (也就是评论表单的代码):

```
1 <!-- Comment Form -->
2 <form id="comment_form" action="" method="post">
3     <h3>Add a comment</h3>
4     <div class="hr dotted clearfix">&nbsp;</div>
5     <ul>
6         <li class="clearfix">
7             <label for="name">Your Name</label>
8             <input id="name" name="name" type="text" />
9         </li>
10        <li class="clearfix">
11            <label for="email">Your Email</label>
12            <input id="email" name="email" type="text" />
13        </li>
```

```
14     <li class="clearfix">
15         <label for="email">Your Website</label>
16         <input id="website" name="website" type="text" />
17     </li>
18     <li class="clearfix">
19         <label for="message">Comment</label>
20         <textarea id="message" name="message" rows="3"
21     cols="40"></textarea>
22     </li>
23     <li class="clearfix">
24         <!-- Add Comment Button -->
25         <a type="submit" class="button medium black right">Add
26     comment</a> </li>
27
28 </ul>
29
30 </form>
```

改成：

```
1 <?php
2 if ( !comments_open() ) :
3 // If registration required and not logged in.
4 elseif ( get_option('comment_registration') && !is_user_logged_in() ) :
5 ?>
```

```
6 <p>你必须 <a href="<?php echo wp_login_url( get_permalink() ); ?>">登录</a> 才能发
7 表评论.</p>
8 <?php else : ?>
9 <!-- Comment Form -->
10 <form id="commentform" name="commentform" action="<?php echo
11 get_option('siteurl'); ?>/wp-comments-post.php" method="post">
12     <h3>发表评论</h3>
13     <div class="hr dotted clearfix">&nbsp;</div>
14     <ul>
15         <?php if ( !is_user_logged_in() ) : ?>
16             <li class="clearfix">
17                 <label for="name">昵称</label>
18                 <input type="text" name="author" id="author" value="<?php echo
19 $comment_author; ?>" size="23" tabindex="1" />
20             </li>
21             <li class="clearfix">
22                 <label for="email">电子邮件</label>
23                 <input type="text" name="email" id="email" value="<?php echo
24 $comment_author_email; ?>" size="23" tabindex="2" />
25             </li>
26             <li class="clearfix">
27                 <label for="email">网址(选填)</label>
```

```
28         <input type="text" name="url" id="url" value="<?php echo
29 $comment_author_url; ?>" size="23" tabindex="3" />
30     </li>
31     <?php else : ?>
32     <li class="clearfix">您已登录:<a href="<?php echo
33 get_option('siteurl'); ?>/wp-admin/profile.php"><?php echo $user_identity; ?> </a>. <a
34 href="<?php echo wp_logout_url(get_permalink()); ?>" title="退出登录">退出
35 &raquo;</a> </li>
36     <?php endif; ?>
37     <li class="clearfix">
38         <label for="message">评论内容</label>
39         <textarea id="message comment" name="comment" tabindex="4"
40 rows="3" cols="40"></textarea>
41     </li>
42     <li class="clearfix">
43         <!-- Add Comment Button -->
44         <a href="javascript:void(0);"
45 onClick="Javascript:document.forms['commentform'].submit()" class="button medium
46 black right">发表评论</a> </li>
47 </ul>
48 <?php comment_id_fields(); ?>
49 <?php do_action('comment_form', $post->ID); ?>
```

```
</form>
```

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

函数名称	函数功能
is_user_logged_in	判断用户是否登录
wp_login_url	博客登录地址
get_comment_author_link	用于获取评论者博客地址
<code>\$comment_author</code>	读取 cookie , 如果该用户之前已经发表过评论则自动帮助用户填写用户名
<code>\$comment_author_email</code>	读取 cookie , 如果该用户之前已经发表过评论则自动帮助用户填写 Email
<code>\$comment_author_url</code>	读取 cookie , 如果该用户之前已经发表过评论则自动帮助用户填写博客地址
<code>do_action('comment_form' , \$post->ID);</code>	该函数为某些插件预留
<code>wp_logout_url</code>	退出登录的链接

好了，评论模板 comments.php 制作完毕！最后本次修改后的主题文件下载：

下载

这个文件

WordPress 主题制作全过程（十一）：制作 page.php

page.php 也就是页面，可能大家不太理解页面是什么意思，博客上的所有网页不都是页面吗？我们这里指的页面一个单独的页面，如"关于"、"联系方式"等，可以在 WordPress

后台 - 页面，进行页面的添加修改等。

这个 page.php 有很多东西跟 single.php 也都是相通的，或者基本都一样，这里我们对部分内容也不做太多的解析，直接上结果，如果有不懂的可以翻翻 [WordPress 主题制作全过程（八）：制作 index.php](#)。

1、添加标题

搜索代码：

```
1 <!-- Column 1 /  
Content -->
```

改成：

```
1 <!-- Column 1 / Content -->  
2 <h2 class="grid_12 caption clearfix"><?php  
the_title(); ?></h2>
```

2、添加页面内容

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel porta erat.  
1 Quisque sit amet risus at odio pellentesque sollicitudin. Proin suscipit molestie facilisis.  
Aenean vel massa magna. Proin nec lacinia augue. Mauris venenatis libero nec odio  
viverra consequat. In hac habitasse platea dictumst.</p>
```

改成：

```
1 <?php  
the_content(); ?>
```

3、添加评论框

将<!-- Contact Form --> 与 </form>之间的所有代码删除 (包括</form>)

替换成：

```
1 <?php  
comments_template(); ?>
```

4、添加查询语句

搜索代码：<!-- Column 1 / Content -->

改成：

```
1 <!-- Column 1 / Content -->  
2 <?php if (have_posts()) : the_post();  
update_post_caches($posts); ?>
```

查找：

```
1 </div>  
2 <?php  
get_sidebar(); ?>
```


改成：

```
1
2     </div>
3
4     <?php else : ?>
5
6     <div class="grid_8">
7         没有找到你想要的页面！
8     </div>
9
10    <?php endif; ?>
11
12    <?php get_sidebar(); ?>
```

我们的 page.php 就制作完毕了。另外还有两个页面，无边栏页面 full_width.php，联系页 contact.php，制作方法完全一样，这里就不再详述了，我已经直接写好了，下载即可。你可能会问，那这两个页面怎么用呢？我分别在这两个页面的前面写了两个注释：

```
1 <?php
2 /*
3  Template Name: Page without
4  sidebar
5  */
6 ?>
7
8 <?php
9 /*
10  Template Name:
```

```
4 contract
```

```
5 */
```

```
?>
```

以后在 WordPress 后台新建页面的时候，可以在右边栏选择页面模板，你可以选择 Page without sidebar 或 contract，不选默认使用 page.php 模板！contact.php 边栏的联系方式你自己改一下内容就可以了。

到目前为止，我们的 WordPress 主题制作系列教程就结束，说实话你也应该没学到啥东西，我讲的实在太差劲了，就当是 WordPress 主题制作的一个感性认识过程吧。最后提供我们制作完成的主题下载，估计不会有人愿意用这样的主题的，太丑了！还有就是评论还没有分页功能。

下载

这个文件