

本文由[万三水煮鱼\(w3123.com\)](http://w3123.com)制作整理，未经允许，请勿以任何形式转载，否则将被视为侵权！

在学习本教程之前我们应该准备些什么？

搭载本地服务器：[一键搭载php+mysql+apache环境教程](#)

如果你想要获得更多教程请登录[万三水煮鱼](#)

[免费的域名和空间](#)信息，在这里为你提供免费的空间信息。

个人觉得wordpress最吸引用户的地方就是它灵活的插件和主题机制,这可以使人们不用任何专业教程便可建立属于自己的博客,即网站.

对于自己的网站你心里是不是也有这样或者那样的想法?网上现在的wordpress主题很丰富，这里不乏优秀的作品，但是是不是和你心里想要的样子总有这样或者那样的不同？毕竟那不是为你自己专门定制的！那就动手制作自己的网站吧！

先打开wordpress主题中的文件，就wordpress默认的主题“twentyeleven”吧。wordpress主题全部存放在“网站根目录\wp-content\themes\”文件夹中，我们打开网站根目录\wp-content\themes\twentyeleven\index.php,这是网站的首页：如下图：

```

1 <?php
2 /**
3  * The main template file.
4  *
5  * This is the most generic template file in a WordPress theme
6  * and one of the two required files for a theme (the other being style.css).
7  * It is used to display a page when nothing more specific matches a query.
8  * E.g., it puts together the home page when no home.php file exists.
9  * Learn more: http://codex.wordpress.org/Template_Hierarchy
10 *
11 * @package WordPress
12 * @subpackage Twenty_Ten
13 * @since Twenty_Ten 1.0
14 */
15
16 get_header(); ?>
17
18 <div id="container">
19     <div id="content" role="main">
20
21         <?php
22         /* Run the loop to output the posts.
23          * If you want to overload this in a child theme then include a file
24          * called loop-index.php and that will be used instead.
25          */
26         get_template_part( 'loop', 'index' );
27         ?>
28     </div><!-- #content -->
29 </div><!-- #container -->
30
31 <?php get_sidebar(); ?>
32 <?php get_footer(); ?>
33

```

其中红框中的内容可以完全省区，那就只剩下十几行英语单词了。

这么简单？

对！就这么简单，你要你认识26个英文字母，你就能学会这篇教程，成为WPer！

就从上图讲起，<?php get\_header(); ?>是php语句，php语句以"<?php"开头，"?>"结尾。而其中get\_header()是wordpress内置函数，所谓内置函数就是不用我们定义就能直接使用的函数。与内置函数对应的自然是外部函数，必须由我们编写定义存于主题目录下"functions.php"才可使用，比如上图中get\_template\_part()就是用户定义函数。由主题twentyeleven定义，在别的主题就不能使用。

其中<div>和</div>用来定义和层，什么是层呢？可以把它想象成一个盒子，盒子里装着网页的一些内容，我们把这些内容比作西瓜。当我们想要把西瓜从左边挪动到右边的时候只需要推动盒子就可以了，而不需要一个一个西瓜的挪动。

如果你没有完全看懂上面一段话，那么不用纠结，直接跳过，因为你不需要完全懂

那些东西。不过懂得这些东西对以后的工作却大有益处，如果你想要了解更多可以参见独立的教程。


有现成的如果可以满足你的要求就用现成的,因为时间比金钱更重要,如果现有要满足你的要求要进行大量的修改,那么就干脆自己再做一个,因为这比对一个毫不了解的东西进行修改要简单的多.

这篇教程说是wordpress主题制作教程，但叫wordpress修改教程，因为这篇教程教你基于现成的主题单单改变它的外观，这会简单很多。

这里我们以修改wordpress默认主题twentyeleven为例，当然你也可以以别的主题为基本。

首先进入后台，确认你当前使用的主题为twentyeleven主题：

当前主题



**Twenty Eleven 1.2 ,**

The 2011 theme for available theme opti transforms your fror custom "Ephemerat custom header imag

打开twentyeleven主题文件夹，很多文件，不用管它，我们修改主题主页。进入后台，选择编辑，我们可以看到其中各个文件的作用。



外观

主题

小工具

菜单

主题选项

背景

顶部

编辑

分类目录模板	(category.php)
图片附件模板	(image.php)
底部	(footer.php)
搜索框	(searchform.php)
搜索结果	(search.php)
文章归档	(archive.php)
文章页面	(single.php)
标签模板	(tag.php)
模板函数	(functions.php)
评论	(comments.php)
边栏	(sidebar.php)
页面模板	(page.php)
顶部	(header.php)
首页模板	(index.php)
样式	
RTL 样式表	(rtl.css)
可视化编辑器样式表	(editor-style.css)
样式表	(style.css)

首页模板，即index.php文件。

打开index.php文件( 不推荐使用微软自带的文本编辑器打开，因为有可能出现乱码这里我使用的是dreamweaver )。打开后直接清空。

在文件中写入以下内容。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

["http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">](http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd)

```
<html xmlns="http://www.w3.org/1999/xhtml">  
  
<head>  
  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
<title>  
  
</title>  
  
</head>  
  
<body>  
  
</body>  
  
</html>
```

保存文档，进入浏览器输入：localhost访问。

我们看到了一个空白的窗口，为什么？因为上面我们只是写了一个HTML基本框架，并没有向这个框架“添加”内容。

简单解释上面代码：

<html> HTML文档开始标签。

<head> 文件头开始标签。

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> 定义文本编码，这里为utf-8；

<title> 定义HTML标题开始标签

</title> 定义HTML标题结束标签

</head> 文件头结束标签。

<body> 文档主体开始标签

</body> 文档主体结束标签

</html>HTML文档结束标签。

如你所见，除了单标签所有HTML开始标签都要有结束标签。形式如</开始标签>;上文中，<meta>标签就是一个单标签。

为文档添加标题

在上面代码中<title> </title>之间添加如下代码：

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

现在的代码如下面：

```
<!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>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>
```

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

```
</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

保存文档，回到浏览器刷新。

现在页面如下图：



我们做了什么？

`<?php bloginfo('name'); ?>` 是一句php语句。前面提过。

其中**bloginfo()**函数是wordpress的内置函数。根据其中的函数输出博客的相应信息。

这句中参数为“name”，所以`<?php bloginfo('name'); ?>`输出的是博客的名字。

### 第三章:

添加博客标题.在`<body>`和`</body>`之间写入以下语句：

`<?php bloginfo('name'); ?>`这个就是前面被用来当页面标题的，前面已经讲过，这里不再多讲。保存文档，刷新浏览器



万三水煮鱼

哈哈网页有东西了，这说明你没有做错。

继续编辑文档。`<?php bloginfo('name'); ?>`后面添加内容：

`<?php bloginfo('description'); ?>`保存文档刷新浏览器。

---

# 万三水煮鱼

又一个 WordPress 站点

我们刚才做了什么？<?php bloginfo('description'); ?>和<?php bloginfo('name'); ?>调用的是同一个函数，只是参数不同罢了。description是描述的意思。所以这句php语句的输出网站的描述。（网站名称和网站描述）在后台设置。我想这个不用说明了吧。。。)

在<?php bloginfo('name'); ?>前添加<h1>;在<?php bloginfo('name'); ?>后添加</h1>。<h1>~<h6>标签用于声明网页内标题

为网站名称添加链接

在<?php bloginfo('name'); ?>前添加内容:<a href="#">;在<?php bloginfo('name'); ?>之后添加内容：</a>

保存文档，刷新浏览器：



[万三水煮鱼](#)

又一个 WordPress 站点

我们可见，网站名称现在已经变成了蓝色，并且可以点击，我们点击，并没有转向别的地址，应为我们还没有添加任何地址，用以下语句代替上句中的#号：

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

现在index.php文档内应该有如下内容：



```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>
6 <?php bloginfo('name'); ?>
7 </title>
8 </head>
9 <body>
10 <a href="<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a><br />
11 <?php bloginfo('description'); ?>
12 </body>
13 </html>

```

最后在<body>之前输入以下内容：

```
<div id="header">
```

在</body>之后添加 </div>

保存文档刷新浏览器。

我们看到页面没有变化。因为我们只是把我们添加的内容“装”在一个叫“header”层里。div是虚拟的层，不可见。它的基本作用我们已经在教程的开始讲过。

#### 第四章：wordpress主循环。

在</body>之前输入以下内容：

```

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

```

保存文档，刷新浏览器。

---

## 万三水煮鱼

又一个 WordPress 站点

页面没有增加任何内容，那我们刚才添加的内容是什么作用呢？

- `if(have_posts())` - 检查博客是否有日志。
- `while(have_posts())` - 如果有日志，那么当博客有日志的时候，执行下面 `the_post()` 这个函数。
- `the_post()` - 调用具体的日志来显示。
- `endwhile;` - 遵照规则 #1，这里用于关闭 `while()`
- `endif;` - 关闭 `if()`
- 注释：并不是所有的代码都需要两部分用来打开和关闭。有些代码能够自我关闭，这就解释了 `have_posts()` 和 `the_post()`；这两个函数。因为 `the_post()`；在 `if()` 和 `while()` 的外面，只需要分号去结束或者关闭。

所以

第五章，调用文章标题。

或许你会想到前面讲到的博客标题调用函数，但这是文章标题和它没有什么关系。

在 `the_post(); ?>` 的后面和 `<?php endwhile; ?>` 的前面输入 `<?php the_title(); ?>`

保存文档，刷新浏览器。



# 万三水煮鱼

又一个 WordPress 站点  
世界，你好！

我们刚才做了什么？

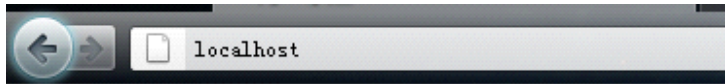
前面提过可知，`<?php the_title(); ?>`是php语句；而title是标题的意思，所以`the_title()`在此用来调取文章标题。

第二节，为文章标题添加链接：

在`<?php the_title(); ?>`输入`<a href="#">`,在`<?php the_title(); ?>`之后添加：`</a>`,保存文档，刷新浏览器。

当我们将鼠标移至文章标题时鼠标变成一个小手，并且可以点击，这说明他已经变成了链接。关于超链接我们前面已经说过。现在的标题依旧是一个空链接。

将`<a href="#">`中的#号替换为以下内容：`<?php the_permalink(); ?>` 保存文档，刷新浏览器。



# 万三水煮鱼

又一个 WordPress 站点

[测试三大件富克拉三大件芬兰空军](#); [世界，你好！](#)

点击文章标题我们来到了文章页面，我们还没有编辑文章页面，所以现在它依旧是默认主题的样子。

现在index.php文档应该有以下内容：

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>
6 <?php bloginfo('name'); ?>
7 </title>
8 </head>
9 <body>
10 <div id="header">
11 <a href="<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a>
12 <?php bloginfo('description'); ?>
13 </div>
14 <?php if(have_posts()) : ?><?php while(have_posts()) : the_post(); ?>
15 <a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
16 <?php endwhile; ?>
17 <?php endif; ?>
18 </body>
19 </html>
```

添加文章内容

在<?php the\_title(); ?> </a>后输入以下内容：：<?php the\_content(); ?>。

保存文档，刷新浏览器。



## 万三水煮鱼

又一个 WordPress 站点

[测试三大件富克拉三大件芬兰空军;](#)

发送颠覆思路打开;发卡斯等;飞撒旦;发酵;送达分;撒旦家;发电量思考激发思考定界符离开家打扫  
[世界，你好！](#)

欢迎使用 WordPress。这是系统自动生成的演示文章。编辑或者删除它，然后开始您的博客！

我们刚才做了什么？

我们依旧是添加了一句php语句。调用了一个叫the\_content()的函数。

the\_content()函数用来调用文章内容。

我是懒人，懒人的集中表现就在于如果可以少些或者不写，能够少说或者少说就不会去累着自己。下面讲解如何添加日志元数据，关于日志元数据的讲解内容大部分截取自“我爱水煮鱼”同样一个由wordpress构建的一个博客。

上面讲解日志元数据的时候偷了懒，也没有让大家用<div>标签去包围日志内容以更容易格式化，因为我们不需要这么复杂。

最后一次保存文档刷新浏览器。然后打开文档，把the\_post(); ?>到<?php endwhile; ?>的内容全部删除。其中不包括这两句php语句。

然后在<?php endwhile; ?>之前写入以下内容：<?php  
get\_template\_part( 'content', 'index' ); ?>

保存文档，刷新浏览器。



## 万三水煮鱼

又一个 WordPress 站点

### [测试三大件富克拉三大件芬兰空军;](#)

发表于 [2011 年 10 月 27 日](#) 由 [xiaopang](#)  
[回复](#)

发送颠覆思路打开;发卡斯等;飞撒旦;发酵;送达分;撒旦家;发电量思考激发思考定界符离开家打扫

发表在 [万三水煮鱼](#) | [发表回复](#) [编辑](#)

### [世界，你好！](#)

发表于 [2011 年 10 月 22 日](#) 由 [xiaopang](#)  
[1](#)

欢迎使用 WordPress。这是系统自动生成的演示文章。编辑或者删除它，然后开始您的博客！

发表在 [未分类](#) | [1 条回复](#) [编辑](#)

我们丢掉的内容又回来了，而原先坑长的代码被这么一句php代码代替。它代替了wordpress主循环的作用。

get\_template\_part()是twentyeleven定义的函数。作用就是调用日志标题，内容和元数据。我们不需要知道它的原理，只需要知道它在这里的用法。

现在的index.php文档应该是这样的。

在<?php if(have\_posts()) : ?>之前添加以下内容:

```
<div id="content">
```

在：<?php endif; ?>之后添加以下内容：

```
</div>
```

保存文档，刷新浏览器。

察看页面源代码：

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>
6 <?php bloginfo('name'); ?>
7 </title>
8 </head>
9 <body>
10 <div id="header">
11 <a href="<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a>
12 <?php bloginfo('description'); ?>
13 </div>
14 <div id="content">
15 <?php if(have_posts()) : ?><?php while(have_posts()) : the_post(); ?>
16 <?php get_template_part( 'content', 'index' ); ?>
17 <?php endwhile; ?>
18 <?php endif; ?>
19 </div>
20 </body>
21 </html>

```

给页面添加侧边栏和底部。

在</body>前面添加如下内容：

```
<div id="sidebar">
```

```
</div>
```

```
<div id="footer">
```

```
</div>
```

在这几句中我们添加了两个id分别是：sidebar和footer的层，但并没有添加任何内容。你也可以给他写些内容，比如这样：

```
<div id="sidebar">
```

```
sidebar
```

```
</div>
```

```
<div id="footer">
```

```
footer
```

```
</div>
```

好吧，为了便于观察，就这样写。

现在index.php的内容：

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>
6 <?php bloginfo('name'); ?>
7 </title>
8 </head>
9 <body>
10 <div id="header">
11 <a href="<?php bloginfo('url'); ?>"><?php bloginfo('name'); ?></a>
12 <?php bloginfo('description'); ?>
13 </div>
14 <div id="content">
15 <?php if(have_posts()) : ?><?php while(have_posts()) : the_post(); ?>
16 <?php get_template_part( 'content', 'index' ); ?>
17 <?php endwhile; ?>
18 <?php endif; ?>
19 </div>
20 <div id="sidebar">
21 sidebar
22 </div>
23 <div id="footer">
24 footer
25 </div>
26 </body>
27 </html>

```

格式化首页。

现在主流的显示器大小是1024/768的，除去滚动条，网页的宽度不能大于1000，否则浏览器底部就会出现滚动条，这对于用户来说是非常不友好的。所以我们要首先规定网页宽度的大小。

首先把网页包裹在一个大的层里。

在<body>之后添加如下内容：

<div id="page">

在<div id="footer">之前添加如下内容：

</div>

大部分格式化主题内容的层叠样式都存放在一个叫style.css的层叠样式表中，所以在我们格式化首页的时候首先要链接样式表。

在<head>标签之后添加如下内容：

<link rel="stylesheet" type="text/css" media="all" href="<?php  
bloginfo( 'stylesheet\_url' ); ?>" />

大家不需要它的作用，但无论在哪个wordpress主题中它都必不可少。

现在打开主题根目录下style.css文件。清空

现在看下index.php文档中的内容，分析下首页的层次分布。

上图就是现在首页源代码的内容。层次很简单也很明确：

所有的网页内容都包括在一个id为page的层里；

id为page的层里包括四个层：id分别为header、content、sidebar、footer，这四个曾分



别是：头部、主体内容、侧边栏、底部。

我们要规定网页宽度为970像素，并且居中显示，这要定义id为page的层来完成，然后其内的四个层已以下内容排列：



页眉即头部，页脚即底部。

现在style.css文件应该是空的，现在向其中写入以下内容：

```
@charset "utf-8";  
/* CSS Document */
```

```
#page{width:970px; margin:auto;}  
#content{width:679px;  
margin-right:21px;  
float:left;  
}
```

```

#sidebar{width:170px;
float:right;
}
#footer{
    width:100%;
    float:left;
    text-align:center;
}

```

现在index.php的内容：

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <link rel="stylesheet" type="text/css" media="all" href="<?php bloginfo( 'stylesheet_url' ); ?>" />
6  <title>
7  <?php bloginfo( 'name' ); ?>
8  </title>
9  </head>
10 <body>
11 <div id="page">
12     <div id="header">
13         <a href="<?php bloginfo( 'url' ); ?>"><h1><?php bloginfo( 'name' ); ?></h1></a>
14         <?php bloginfo( 'description' ); ?>
15     </div>
16     <div id="content">
17         <?php if( have_posts() ) : ?><?php while( have_posts() ) : the_post(); ?>
18         <?php get_template_part( 'content', 'index' ); ?>
19         <?php endwhile; ?>
20         <?php endif; ?>
21     </div>
22     <div id="sidebar">
23         sidebar
24     </div>
25 </div>
26 <div id="footer">
27     footer
28 </div>
29 </body>
30 </html>

```

到现在为止我们已经完成了首页的制作，但一个完整的wordpress网站并不是只有一个首页。我们还有许多文件需要修改，比如，存档页面（）、分类目录页面（）、等等，不过好在这些页面和主页没多些什么。都是头部、文章主体、侧边栏和底部。这东西中头部、侧边栏、底部都和首页显示的相同，我们不需要修改可以直接调用，不过在直接调用之前我们还把他们分离为单独的文档，按下面的步骤做：

打开主题目录下：header.php文件，清空原内容。

剪切index.php文件内<div id="content">前所有内容粘贴至header.php文件，保存header.php文件，关闭header.php文件。

打开footer.php文件。清空原内容。

剪切从<div id="footer">开始到结尾的所有内容粘贴至footer.php文件内。保存footer.php文件，关闭footer.php文件。 — — — —

打开sidebar.php文件。清空原内容。

剪切从<div id="sidebar">开始到结尾的所有内容粘贴至sidebar.php文件内。保存sidebar.php文件，关闭sidebar.php文件。

现在index.php剩下的内容：

我们剪贴了index.php文件中的内容。所以就要重新引用。

在文档的起始部分添加如下语句：

```
<?php include('header.php'); ?>
```

在文档的末位加入以下语句：

```
<?php include('sidebar.php'); ?>
```

```
<?php include('footer.php'); ?>
```

保存文档，刷新浏览器，看看首页是不是没有缺少任何内容？

注：include()函数既不是wordpress内建函数也不是主题定义函数，而是php语言内部函数。所以也可以直接使用。他的作用就是加载制定文件。

到现在为止,我们的首页基本已经完成.下面就是一些子页面的修改,前边说过,这些页面和首页并没有大的出入.

还记得前面那个代替wordpress主循环的函数吧。<?php  
get\_template\_part( 'content', 'index' ); ?>，在这个语句中content和index是get\_template\_part()函数的参数。index是表明是在首页模板使用它的。这个函数的作用让我想起了wordpress内建的函数the\_title()函数。  
这个函数在分类页面输出分类标题；在文章页面输入文章标题；在单个页面输入页面名字。神奇的函数。

打开index.php文档，用get\_template\_part()代替  
get\_template\_part( 'content', 'index' )刷新浏览器，输出正常。那么  
get\_template\_part('content', 'index' )是不是能够在首页自动输出最新文章，在分类页面输出该分类下最新文章，在作者页面可以只输出该作者的文章，在月份页面自动输出该月发表的文章呢？经作者测试完全是这样的。所以现在复制index.php内的内容到替换原内容就可以了。 — — —

### 窗体化侧边栏

打开sidebar.php文件，删除我们原先乱写的文字，那么此文件现在应只剩下如下内容：

```
<div id="sidebar">  
</div>
```

在<div id="sidebar">之后加入如下内容：

```
<?php if(function_exists('dynamic_sidebar')&&dynamic_sidebar()):else: ?>  
<?php endif; ?>
```

现在窗体化侧边栏我们就已经完成了，而上面的代码我们不需要理解。