

wordpress theme in action

第一版



冰封

目录

前言..... 3

第一章：工具篇..... 5

 本地运行环境..... 5

 浏览器..... 6

 文本编辑器..... 8

 图像处理软件..... 8

第二章：预备篇..... 9

 设计草稿..... 9

 一些说明..... 11

第三章：基础篇..... 12

 开胃菜..... 12

 Header.php..... 16

 Index.php..... 23

 Searchform.php..... 32

 Functions.php..... 32

 Sidebar.php..... 32

 Single.php..... 39

 Page.php..... 39

 Comments.php..... 40

 Footer.php..... 47

 404.php..... 52

 screenshot.png..... 53

第四章：进阶篇..... 54

 设置模板..... 54

 使用 jQuery..... 56

 显示相关日志..... 61

 显示最新文章..... 62

 显示特定分类的文章..... 62

 随机顺序显示日志..... 62

 不同的节日显示不同的消息..... 62

 一些 css 的小技巧..... 63

附录 A: 常用函数*..... 68

 Wordpress Header 头部 PHP 代码..... 68

 Wordpress 主体模板 PHP 代码..... 69

 其它的一些 Wordpress 模板代码..... 71

前言

最早接触博客应该是 02 年进入大学以后，当时对于网上的新事物都很好奇，于是在博客中国注册了一个帐号，可惜只发了一篇水文，就一直晾在那里。这样一直过了两年多，到了 05 年的时候，和好友聊 msn，发现她在 msn space 上的博客设计的很精美，而她也鼓励我在 msn 上写写博客。于是又燃起了我的热情，开通了个 space，从网上找了很多教程，特别是在 atiger 这个网站上，学到了很多关于 msn space 设置的技巧。这样在 space 上写了大半年，渐渐的，我实在忍受不了 space 对浏览器的摧残，反应极慢，而且经常假死，于是放弃了这块阵地，从而转向了其他的 BSP，先后用过和讯，新浪，百度，网易，西湖，歪酷等。一个偶然的的机会，让我遇到了 wordpress，顿时让我有种相见恨晚的感觉，从此便爱上了她。后来慢慢的去了解她，在了解的过程中越发喜爱她，我很喜欢扩展性好，自定义功能强的软件，比如 firefox，emacs 等。而 wordpress 也正是这样，她有无数的插件，主题，可供挑选，如果还不满足，可以自己动手，设计自己满意的界面，撰写符合自身需求的插件。而且 wordpress 是开源的，遵循 GPL 协议，这就意味着你不需要花一分钱，就可以得到一个相当强大的博客系统，甚至将她改造成一个 cms。wordpress 中文论坛是一个不错的讨论社区，虽然现在里面挤满了收费空间代理人士，但这并不让我讨厌这个论坛。每当看到一些朋友咨询一些非常简单的问题时，我就想写一篇关于主题制作方面的文章，用实例来讲解如何制作主题，故写了此文。水平有限，而且时间仓促，难免出现一些错误，请多多包涵，也欢迎来信指正。

本书字体

时尚中黑简体

各级标题

Courier

示例代码

雅黑

正文

适合谁阅读？

1. 刚接触 wordpress，想要制作出自己的第一个主题的人
2. 已经厌烦了使用大众化的主题，需要在 web2.0 时代彰显个性的人

不适合谁阅读？

1. 从未接触过 wordpress，以后也不想使用 wordpress 的人
2. 对制作 wordpress 主题已经很熟悉的人

欢迎您提出宝贵的意见

martz (冰封)

Email: moon.zju@gmail.com

版权声明

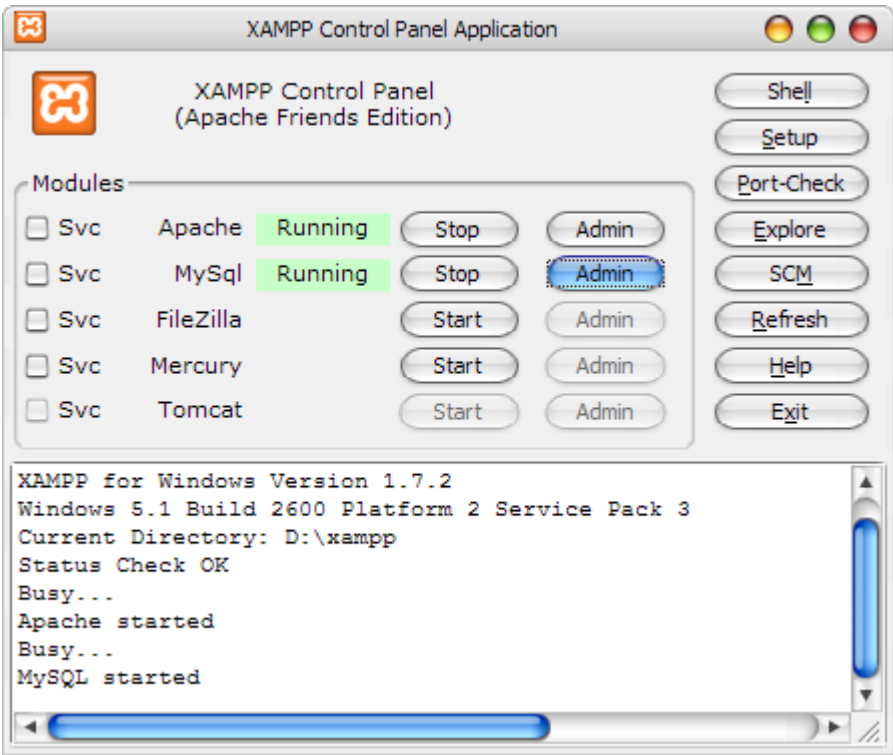
本文可自由下载，阅读，分发，但禁止用于任何商业用途。

第一章：工具篇

俗话说，工欲善其事，必先利其器。在制作主题前，必须先将工具准备齐全，这样制作起来才得心应手，事半功倍。接下来简单说明一下必要的工具。

本地运行环境

在制作主题时需要实时预览主题，在远端主机上测试及预览主题并不是很明智。首先，每次修改了主题，你都需要重新上传修改过的文件，当然你可以直接在 wp 的后台修改主题的代码，不过后台的编辑功能比较差，比如没有语法高亮，没有自动缩进等，这点在后面还会说到。另外，如果你突然想增加一张背景图片，你就需要用 ftp 上传到主题的目录，那是很麻烦的事，所以需要有一个本地的测试环境。网络上有很多集成了 apache , php , mysql 的软件，比如 easyphp , WAMPServer2。这些软件免去了配置 apache , php , mysql 的过程，让你将注意力专注于正事上。我个人推荐一下 xampp ， 这是图形界面的控制面板



默认 mysql 的 root 密码是空的，所以在你的 config.php 文件里不必填写密码。

浏览器

在制作主题的时候，需要实时进行预览，一个浏览器必不可少，当然，一个并不够，因为要兼顾兼容性的问题，在这个浏览器里看上去很正常的主题，到了另一个浏览器里显示却出了大问题，所以至少需要准备 2 到 3 个不同的浏览器来做测试。我把 firefox 作为主力浏览器而不是 ie，因为 firefox 是比较符合 w3c 标准的浏览器，而 ie 是很不规范的浏览器（到了 ie7，ie8 似乎好了一点），另外你有精力，也可以准备个 opera 或者 chrome 等。如果要查看主题在不同 ie 版本下的效果，安装一个 ietest 软件似乎是个不错的方法。使用 firefox 除了标准性之外，还有一个好处就是它有很多扩展可供使用，可以随时扩充浏览器的功能，这里推荐一些对制作主题很有帮助的扩展：

1. Firebug：一个相当强大的 firefox 扩展，是 web 开发者最爱的扩展之一，下图演示的是将鼠标移到源码区，就可以看到对应的区块，以及它的 margin，padding，用不同的颜色示例，这样你就能直观的了解一个 html 元素占用的空间，以及它的外边距和内补白了。



2. Colorzilla : 一个取色器，方便查看网页上的颜色对应的值



3. IE tab : 可以在 firefox 里调用 ie , 这样就不必新开一个 ie 浏览器了。

文本编辑器

很难想象，有很多人就是用 windows 自带的记事本来写代码的。说实话，notepad 真是一个一无是处的编辑器（请原谅我的过激言论），不能显示 linux 换行（因为换行符会变成一个实心方块），没有语法高亮，没有自动缩进，没有.....，几乎可以说什么都没有，而且保存成 utf8 文档时还会自动加上 bom。语法高亮不但让你可以清楚的看到代码的结构，比如你可以一清二楚的区分关键字和字符串，而且让你知道什么时候你把关键字或者函数名打错了（因为没有高亮了）。自动缩进也很方便，省去了你按空格，注意，用 tab 来缩进并不是很好，一般来讲，一个 tab 占 8 个空格的位置，但是 8 个空格的长度来缩进 html 或者 php 实在太长了，而且像那些低级的编辑器（比如记事本），没法更改 tab 的宽度。即使你的编辑器可以更改 tab 宽度，但是不同的平台，不同的编辑器 tab 宽度设置可能不一样，这样就没有一致的缩进风格。混用 tab 和空格是更加不好的编程习惯，推荐你只用空格来缩进，至于宽度，2 个空格或者 4 个空格是个不错的选择。我不打算说教，说服你使用 vim 或者 emacs 这样学习曲线比较陡的编辑器，也不打算推荐你使用 eclipse 或 netbeans 这样强大而耗资源的 ide，但至少你应该把 notepad 扔掉，换成 notepad++，editplus，emeditor，ultraedit 之类的编辑器。

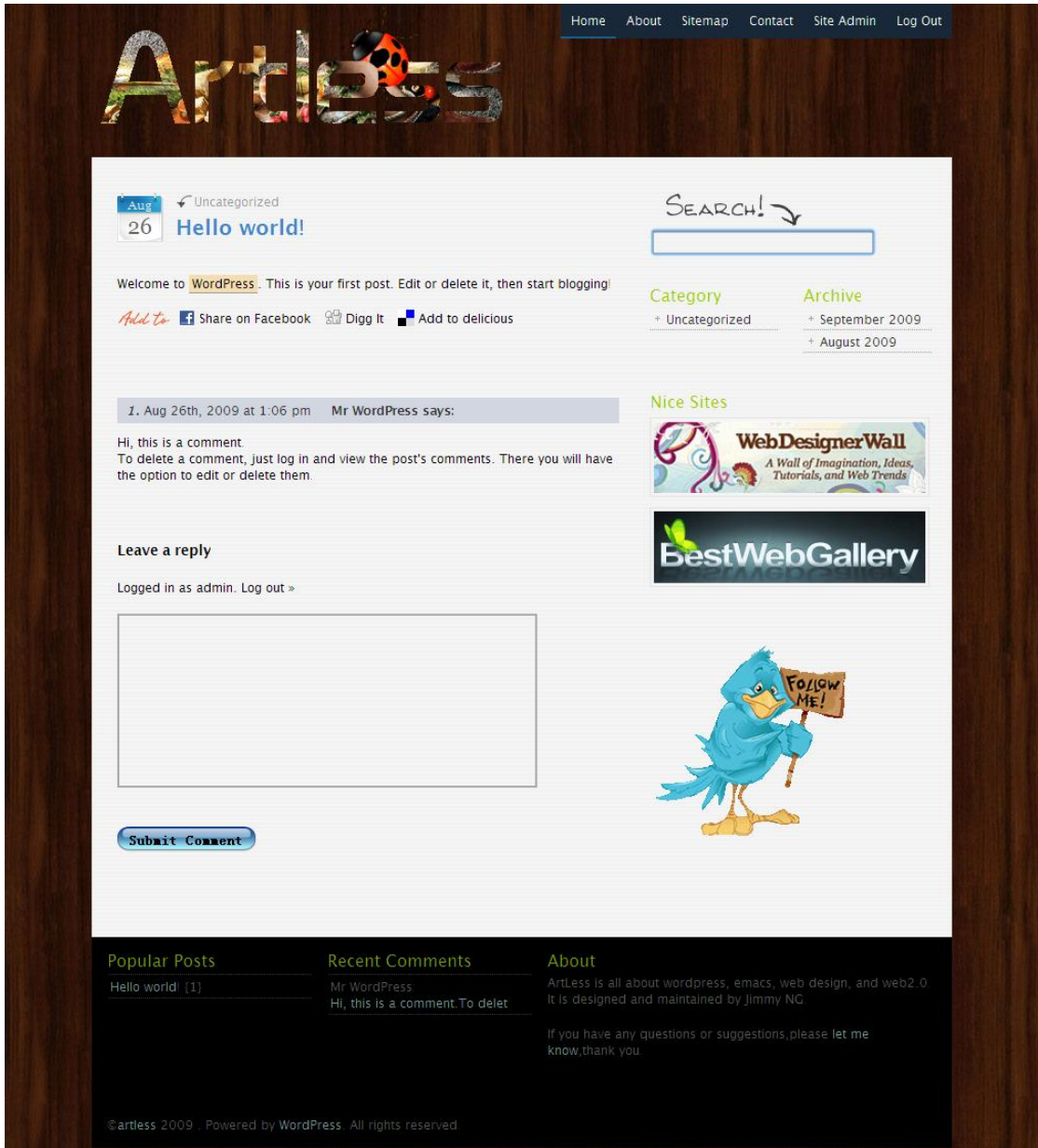
图像处理软件

这个不是必须的，简洁的主题可以只用 css 来美化，不过有时候还是需要用到这类软件，比如制作一个个性化的 logo。图像处理软件，毋庸置疑首推 photoshop，不过为了制作一个 logo 或者一张背景图片似乎杀鸡用牛刀了，这个各位自己看着办吧。

第二章：预备篇

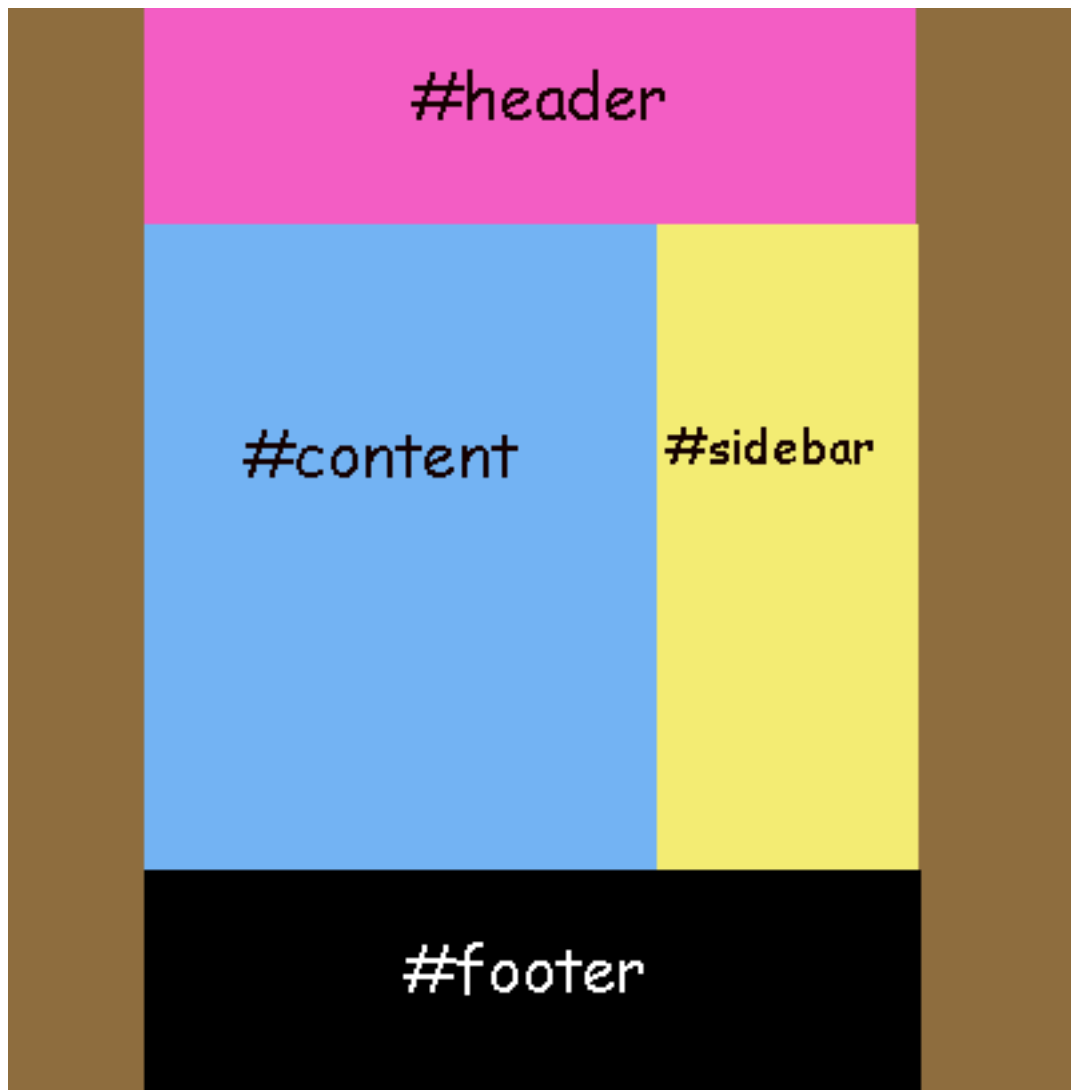
设计草稿

首先预览一下制作完成的主题的效果图



在制作主题之前，先要问一下自己，要制作一个怎么样的主题，是做一个单栏式的，双栏式的还是三栏式的，是浅色调的还是深色调的，要放一些什么内容到主题上，是简洁的以 css 样化为主还是用大量图片来美化，等等。然后画出草图，你可以画在你的脑海里，可以画在草稿纸上，当然也可以画在电脑上。

这里有一张抽象化之后的草图，



这个草图定义了基本的结构框架，从中可以看出这是一个双栏式的设计。以及各个部分的位置，这样就方便我们用 css+div 来布局了。

一些说明

本文省去了一些关于 html 和 css 的介绍，因为这是一篇介绍主题制作的文章，而不是一篇 html 或者 css 的教程，但是在继续下面的章节之前，想做一些关于 html 的说明。

文本用的是 xhtml，所以和 html 有一些区别，比较重要的如下：

1. 所有的标签都要小写

Html 是不区分大小写的，你可以把<html>写成<HTML>，但是 xhtml 其区分大小写的，所以所有的标签都要小写，
这样的标签是不合法的

2. 所有的标签都要闭合

Html 里面标签可以不必和，比如

```
<p>hello world
```

是合法的，但是在 xhtml 里，一定要写成

```
<p>hello world</p>
```

对于没有成对出现的标签，比如
，，直接在最后加/来闭合，比如

```
<br />  <hr />
```

3. 标签要合理嵌套

如果把标签看成是一个一个箱子的话，只能大箱子里装小箱子，你不能戳破一个箱子，使箱子的一部分属于内部，一部分属于外部，比如下面这样是不合法的

```
<p>hello <em>world</p></em>
```

你应该写成

```
<p>hello <em>world</em></p>
```

4. 所有的属性都要加引号

Html 里这样写是合法的

```
<p class=foo>hello world</p>
```

但是在 xhtml 里，你应该写成

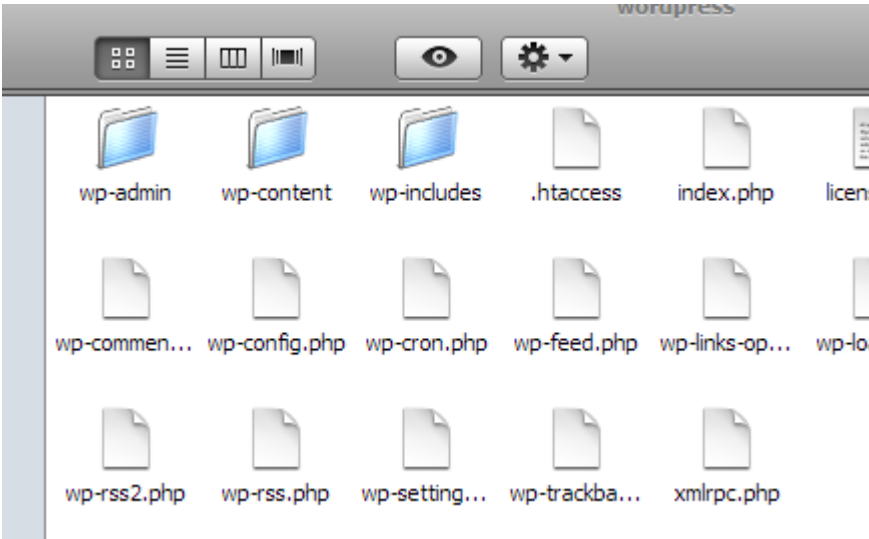
```
<p class="foo">hello world</p>
```

第三章：基础篇

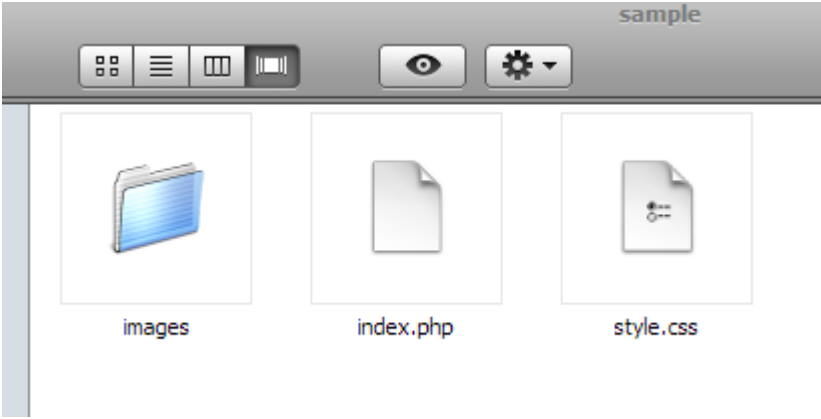
开胃菜

Xampp 目录下的 htdocs 目录就是本地网站的根目录，解压 wordpress 到这里，修改一下 wp-config.php 文件里的数据库用户名，密码，数据库名，就可以安装了，安装过程很快，只需要几秒钟，这里不再累述。

Wordpress 目录有三个子目录，分别是 wp-admin，wp-content，wp-includes，如下图所示



而 wp-content 目录下的 themes 目录，就是存放 wordpress 主题的地方了，其下有很多子目录，每一个目录就是一个主题。在制作之前，先在 themes 目录下新建一个文件夹，名字可以随便取，这里取名为 sample 作为示例，然后在 sample 目录里新建一个叫 images 的文件夹，以后所有主题要用到的图片都会放到这里，然后新建两个文件，一个是 index.php，另一个是 style.css（注意，如果你要制作中文主题，请务必将文件编码改成不带 bom 的 utf-8，如果是英文主题，这个不是必须的），经过这些步骤，sample 文件夹看起来应该是这样的：



启用这个主题，在浏览器里看看是什么样子？

或许你很惊讶，的确，在浏览器里观看，是一片空白，因为这里只建了一个空白的 index.php，当然不会显示任何东西。然后，在 index.php 加入以下代码

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title><?php bloginfo('name'); ?></title>
  </head>
  <body>
    <?php echo 'Hello world' ?>
  </body>
</html>
```

刷新一下，看看会有什么结果？

这是你应该看到显示了 Hello World，先来分析一下以上的代码：

第一行和第二行是 doctype 声明，你不用去理会这是什么，只要把它放到 html 标签以前就可以了，<html>到</html>之间的内容就是全部 html 的内容，浏览器会忽视</html>标签之后的所有东西，而显示在浏览器上则是<body>和</body>之间的内容。从上面来看，只有一句话

```
<?php echo 'Hello world' ?>
```

这就是一句典型的 php 语句，它以<?php 开头，以?>结束，服务器上的 php 解析器会解析所有<?php 和?>之间的内容，而忽略外面的内容，解析完后从服务器传送过来的实际内容其实是（这里省略了 http 头的内容）

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title>你的博客名字</title>
  </head>
  <body>
    Hello World
  </body>
</html>

```

然后浏览器解析 html 的标签，把内容呈现给我们，所以我们就在浏览器里看到了

Hello World 这两个单词，这就是全部的过程，你了解了这个过程之后，我们可以正式迈开大步了。

在第二章已经规划好了各个组成的部分，接下来将 index.php 更新为

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>"
    type="text/css" media="screen" />
    <title><?php bloginfo('name'); ?> </title>
  </head>
  <body>
    <div id="wrap">
      <div id="header"></div>
      <div id="main">
        <div id="content"></div>
        <div id="sidebar"></div>
      </div>
      <div id="footer"></div>
    </div>
  </body>
</html>

```

细心的你一定会发现多了很多东西，首先在<head></head>标签之间引入了一个外部样式表，就是之前在 sample 文件夹里新建的那个 style.css 文件。另外就是在<body></body>之间加了很多 div。什么是 div？不妨把 div 看成是一个个透明的箱子，你可以给每一个 div 取个名字，如<div id="xxx"></div> 或者<div class="xxx"></div>，xxx 就是

箱子的名字, id 和 class 的唯一区别就是, 带 id 的 div, 在同一个页面里只能出现一次, 而带 class 的 div 却没有这个限制。上面这段代码新建了很多空的箱子, 最外面的叫做 wrap 的箱子, 里面包裹了三个箱子, 分别是 header, main, footer, 而 main 这个箱子里又包裹了两个更小的箱子, 叫做 content 和 sidebar。因为是空的, 所以在浏览器里不会出现任何内容。到这里, 就已经完成了主体的框架, 接下来要添砖加瓦了。你肯定不想你的博客只显示一个空白页面吧? !

在添加新代码之前, 先来分割一下文件, 把 index.php 里的内容按下图分别放到这四个不同的文件里

header.php
index.php
sidebar.php
footer.php

如图所示

The screenshot shows an Emacs editor window titled 'emacs@GNU'. The code is divided into four sections, each corresponding to a file:

- header.php**: Contains the DOCTYPE declaration, HTML head section (including a link to a stylesheet and a title), and the opening of the body and wrap divs.
- index.php**: Contains the opening of the main div and the content div.
- sidebar.php**: Contains the closing of the content div and the opening of the sidebar div.
- footer.php**: Contains the closing of the sidebar div, the footer div, and the closing of the body and HTML tags.

The status bar at the bottom indicates the current file is 'c** - *scratch*' and the buffer is 'All (12,0) (XHTML yas Abbrev Fill)'. The code is color-coded: blue for tags, green for attributes, and red for text.

将代码分别放到 4 个文件中, 保存。这里有一个重要的步骤, 请注意

分割完后，index.php 文件里应该只有这么两行

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

然后改成如下

```
<?php get_header(); ?>
    <div id="main">
        <div id="content"></div>
    <?php get_sidebar(); ?>
    <?php get_footer(); ?>
```

和原来有什么区别吗？多了三行，这三行分别是用来导入 header.php, sidebar.php 和 footer.php 的，因为这样才构成一个完整的 html 文档。

Header.php

打开 header.php

在<div id="header"></div>之间添加以下代码，完成后应该是这个样子的

```
<div id="header">
    <div id="logo">
        <a href="<?php echo get_option('home'); ?>/">
            <?php bloginfo('name'); ?>
        </a>
    </div>
    <ul>
        <?php wp_list_pages('title_li=&depth=1&sortcolumn=menu_order'); ?>
    </ul>
</div>
```

刷新后看上去应该是这样的



[artless](#)

- [About](#)
- [sitemap](#)
- [contact](#)

页面上出现了你博客的名字的链接，还有一个页面（page）的列表，用 firebug 查看一下，可以看到 header 部分包括的范围。



很丑，是不是？因为样式表是空的，所以显示出来是最原始的界面，丝毫没有经过人工的修饰，用文本编辑器打开 style.css，加入

```
*{
  margin:0;
  padding:0;
}
```

```
body {
```

```
font:normal 12px/16px 'Lucida Grande', Verdana, Arial,
Sans-Serif;
background: url(images/wood-bg.jpg);
color: #000;
}

a{
outline:none;
text-decoration:none;
color:#000;
}

p {
padding: 0 0 1.5em;
}

ul,ol{
list-style:none;
}

#wrap{
width:840px;
margin:0 auto ;
text-align:left;
}

#header{
width:840px;
height:150px;
position:relative;
}

#logo{
text-indent:-999em;
position:absolute;
left:0;
top:0;
}

#logo a{
display:block;
width:420px;
height:125px;
background:url(images/logo.gif) no-repeat;
```

```
}

#header ul{
  position:absolute;
  right:0;
  top:0;
  text-transform:capitalize;
}

#header ul li{
  float:left;
  background:#192839;
}

#header ul li a{
  display:block;
  padding:8px 10px;
  color:#fff;
  border-bottom:2px solid #192839;
}
```

下面来解释一下这些代码的意义。

```
*{
  margin:0;
  padding:0;
}
```

首先，用*这个通配符来消除所有元素的 margin 和 padding，因为各个浏览器对 html 元素都有自己默认的 margin 和 padding，同一种 html 元素在不同的浏览器里会有不同的 margin 和 padding，这样同一个页面在不同的浏览器浏览时会产生细微的差异，所以先要消除边界和补白，把它们设为 0。

```
body {
  font:normal 12px/16px 'Lucida Grande', Verdana, Arial,
  Sans-Serif;
  background: url(images/wood-bg.jpg);
  color: #000;
}
```

Body 部分设置了一下字体，大小，行高，前景色，背景图片等。

```
a{
  outline:none;
  text-decoration:none;
  color:#000;
}
p {
  padding: 0 0 1.5em;
}
ul,ol{
  list-style:none;
}
```

消除链接的下划线，外轮廓，同时将颜色设为黑色。段落之间设置一些间隔，这里将每个段落的下空白设成 18px(1.5em=1.5×12px)，同时，对于列表，不要让它产生难看的圆点。

```
#wrap{
  width:840px;
  margin:0 auto ;
  text-align:left;
}
```

接下来是#wrap，还记得它吗？它就是那个大箱子，#代表这是 id，而不是 class，wrap 就是名字 将它设置成 840px 宽，为了让它居中显示，设置 margin:0 auto;这在 firefox 等标准浏览器里工作的很好，但是 IE6 却仍旧我行我素，并不居中。为了让它在 IE6 里也居中显示，需要在它的父容器，就是 body 里设置 text-align:center;

接下来就是 header 部分的所有样式化命令了，还记得 header 这个箱子里有什么吗？对了，放置了一个博客首页的链接和一个页面的列表。

链接包裹在 id 为 logo 的箱子里

```
#logo{
  text-indent:-999em;
  position:absolute;
  left:0;
  top:0;
```

```
}
```

第一行设置一个很大的负缩进，这样链接的文字就不会显示在页面上了，然后将位置设置为绝对定位，把它定位到最靠左上角（因为 left 是 0，top 也是 0），注意，这样做先要在父容器设置 position: relative; 这样才会对父容器进行定位，不然会以 body 为参照，这就不是我们想要看到的结果了。Logo 的父容器是什么？当然是 #header 了。

```
#logo a{
  display: block;
  width: 420px;
  height: 125px;
  background: url(images/logo.gif) no-repeat;
}
```

首先，将原本是内联元素的链接设置成 display: block; 这样才能设置宽度和高度，而这个宽度和高度就是 logo 图片的大小，最后把 logo 图片设置成背景。

```
#header ul{
  position: absolute;
  right: 0;
  top: 0;
  text-transform: capitalize;
}
```

同理，对无序列表也进行了绝对定位，把它放到了最右上角，并且设置每个单词首字母大写。

```
#header ul li{
  float: left;
  background: #192839;
}
```

对无序列表中的列表元素进行浮动，这样列表元素就可以横排了，设置一下背景色。

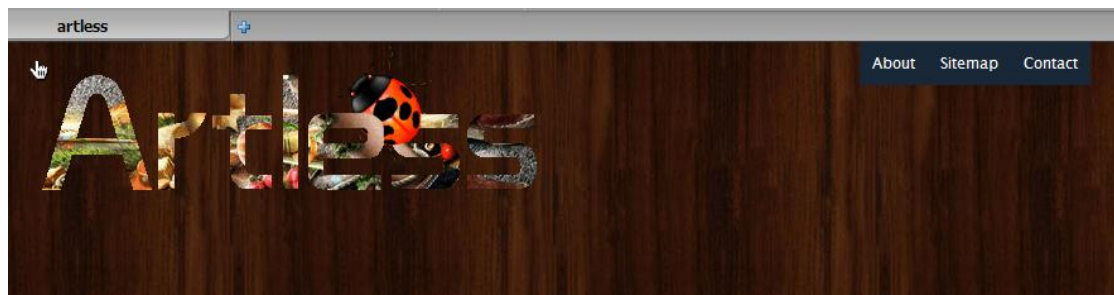
```
#header ul li a{
  display: block;
  padding: 8px 10px;
  color: #fff;
  border-bottom: 2px solid #192839;
}
```

同理，这里也设置了 display: block; 并设置了 2px 宽的下边框

```
#header ul li a: hover{
  border-bottom: 2px solid #4082ae;
}
```

当鼠标移动到列表元素上时，下边框的颜色会改变，增加互动性。

经过以上设置，重新刷新你的浏览器，看看是怎么样



一个 logo 图片替代了难看的文字链接，列表也好看了很多。

这样 header 就完全设置好了？不，没有，我们只设置好了我们说看到的部分，还有我们从浏览器上看不到的部分（<head></head>之间）还基本上没有设置

打开 header.php，看一下 body 标签之前的内容

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>"
    type="text/css"
    media="screen" />
    <title><?php bloginfo('name'); ?> </title>
  </head>
```

这里只是简单的设置了一个 title 和引入了一个外部样式表而已，接下去要扩充它

```
<!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"
  <?php language_attributes(); ?>>
  <head profile="http://gmpg.org/xfn/11">
    <meta http-equiv="Content-Type"
    content="<?php bloginfo('html_type'); ?>;
```

```

charset=<?php bloginfo('charset'); ?>" />
    <title><?php wp_title('-', true, 'right');?>
<?php bloginfo('name');?>
    </title>
    <link rel="stylesheet" type="text/css" media="screen"
href="<?php bloginfo('stylesheet_url'); ?>" />
    <link rel="alternate" type="application/rss+xml"
title="<?php bloginfo('name'); ?> RSS Feed"
href="<?php bloginfo('rss2_url'); ?>" />
    <link rel="alternate" type="application/atom+xml"
title="<?php bloginfo('name'); ?> Atom Feed"
href="<?php bloginfo('atom_url'); ?>" />
    <link rel="pingback" href="<?php
bloginfo('pingback_url'); ?>" />
    <link rel="shortcut icon" href="<?php
bloginfo('template_url'); ?>/favicon.ico" type="image/x-icon" />
    <script type="text/javascript" src="<?php
bloginfo('template_url'); ?>/js/jquery-1.3.2.min.js"> </script>
    <script type="text/javascript" src="<?php
bloginfo('template_url'); ?>/js/global.js"></script>
    <?php wp_head(); ?>
</head>

```

到目前为止，只要照搬上面的代码就可以了。

Index.php

这是重头戏，写博客，当然要显示日志，显示不了日志，就失去了博客的意义了。

把所有的日志(posts)都放在在 content 这个箱子中 ,在<div id="content"></div>

之间加入

```

<?php if (have_posts()) : ?>
    <?php while (have_posts()) : the_post(); ?>
        <div <?php post_class() ?> id="post-<?php the_ID(); ?>">
            <h2><a href="<?php the_permalink();?>" rel="bookmark">
<?php the_title(); ?></a></h2>
            <div class="entry">
                <?php the_content('Read more'); ?>
            </div>
        </div>
    <?php endwhile; ?>
<?php else : ?>
    <div class="post">
        <h2>Not Found</h2>
        <div class="enrty">
            <p>Sorry, no post yet!</p>
        </div>
    </div>
<?php endif; ?>

```

保存完后你刷新浏览器，应该可以看见日志显示出来了，在样式化 content 部分之前，先来解释一下这段代码的意思。

这段代码这就是一个所谓的 loop，也就是一个循环，你可能会在网上看到一些 wordpress 的小技巧，会告诉你把代码放到哪里，有些是放到 loop 之前，有些则要放到 loop 当中。其实 loop 就是个 while 循环而已。

```

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

```

这段代码检测是否有日志，如果有日志，have_posts() 会返回 1，代表 true，就会执行接下来的 loop，否则就执行 else 部分的内容，从上面可以看出，如果没有日志的话，会显示 sorry, no post yet!


```

<?php while (have_posts()) : the_post(); ?>
    <div <?php post_class() ?> id="post-<?php the_ID(); ?>">
        <h2><a href="<?php the_permalink(); ?>" rel="bookmark">
<?php the_title(); ?></a></h2>
        <div class="entry">
            <?php the_content('Read more'); ?>
        </div>
    </div>
<?php endwhile; ?>

```

这是一个 while 循环,通过循环,会将日志按照时间顺序(越新的日志越靠前)一篇一篇的显示出来。

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

会显示标题的永久链接地址

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

会显示标题

```
<?php the_content('Read more'); ?>
```

会显示文章的内容,如果在文章中设置了<!--more-->

则会被截断,显示 Read more 替代后面的内容,你可以任意修改引号里的字符串。比如改成 '阅读更多'

Ok,是该样式化日志部分的时候了,打开 style.css

```

#main{
    background:url(images/main-bg.gif);
    text-align:left;
    width:840px;
}

```

main 箱子包括了 content 和 sidebar,设置背景 main-bg.gif,文字左对齐,宽度为 840px

```
#content{
  width:490px;
  overflow:hidden;
  float:left;
  margin:35px 0 35px 25px;
  display:inline;
}
```

content 就是用来包裹日志的箱子,设置宽度为 490px 如果里面的内容宽度超过 490px,自动截断,这样就不会撑破这个箱子或者出现一个滚动条了,然后设置左浮动,设置一下上下左右的边距,因为 IE6 有一个两倍边距的 bug,所以要设置 display:inline;来修正这个 bug,这种技巧就是人们常说的 css hacks

```
.post{
  padding:0 0 30px 0;
}
```

这是设置日志与日志之间的距离,不然日志间就靠的太紧了

```
.post h2{
  color:#4488cc;
  font-size:20px;
}
.post h2 a{
  color:#4488cc;
}
```

```
.post h2 a:hover{
  color:#000;
}
```

设置日志标题的颜色,字体大小,以及鼠标移动到标题上字体的颜色

```
.entry{
  padding:30px 0;
  text-align:justify;
  clear:both;
}
```

设置正文和标题之间的距离,并清除左右浮动

刷新一下网页，看看效果怎么样？



咦，刚才明明给 wrap 这个箱子设置了背景图片，为什么没有显示出来？

原因出在 content 上，因为对 content 设置了浮动，对父元素来说，浮动的元素没有高度，所以 wrap 不能根据 content 的高度来调节自己的高度，因为 wrap 的高度是 0，当然看不到背景图片啦？如何解决这个问题？

解决方法就是在 wrap 箱子里的最后一个浮动元素之后，清除掉所有浮动

打开 sidebar.php，这时只有一行

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

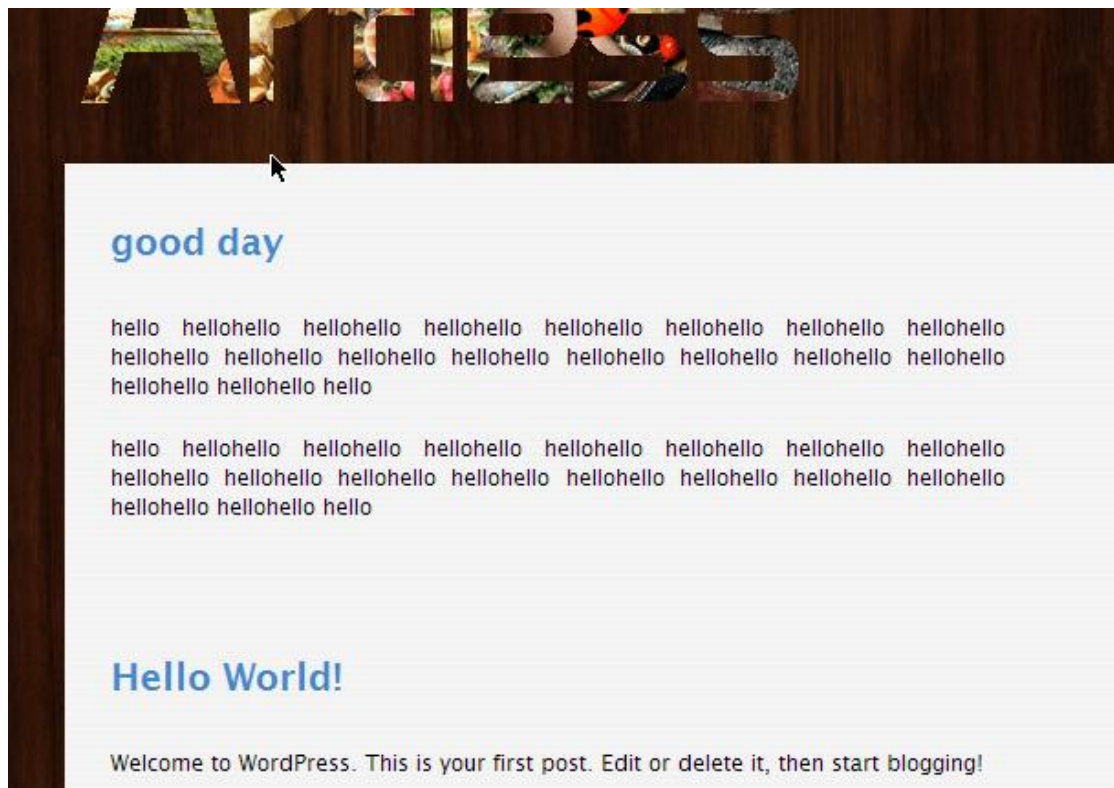
在其后添加

```
<div class="clear"></div>
```

然后在 css 设置

```
.clear{
  clear:both;
}
```

重新刷新一下页面



一切正常了 不过看上去似乎有些单调 因为只显示了一个标题 还有正文 其实wordpress 还可以显示一些其他东西，比如分类，作者，发表时间，评论数目等等

```
<?php the_author() ?>
```

显示作者

```
<?php the_category(',') ?>
```

显示分类，中间用逗号隔开，你也可以换成其他的符号来分割

```
<?php comments_popup_link('0', '1', '%'); ?>
```

显示评论的数目，并链接到评论，你可以修改字符串，比如没有评论，1 条评论，%条评论

```
<?php the_time('F jS, Y') ?>
```

显示日志发表的时间，参数说明如下

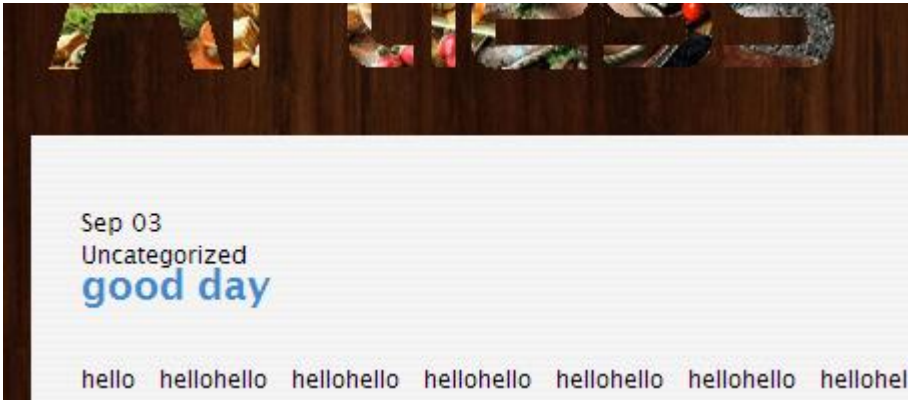
字符	说明	例子
d	月份中的第几天，有前导零的 2 位数字	01 到 31
D	星期中的第几天，文本表示，3 个字母	Mon 到 Sun
j	月份中的第几天，没有前导零	1 到 31
l	星期几，完整的文本格式	Sunday 到 Saturday
S	每月天数后面的英文后缀， st, nd, rd 或者 th	可和 j 一起用
F	月份，完整的文本格式	January 到 December
m	数字表示的月份，有前导零	01 到 12
M	三个字母缩写表示的月份	Jan 到 Dec
n	数字表示的月份，没有前导零	1 到 12
Y	4 位数字完整表示的年份	1999, 2006
y	2 位数字表示的年份	06

由于本主题只想显示日期和分类，故省去了其他内容

将原先的 h2 删去，改成以下内容

```
<div class="date">
<span><?php the_time('M') ?></span><?php the_time('d') ?>
</div>
<div class="title">
  <div class="category"><?php the_category(', ') ?></div>
  <h2><a href="<?php the_permalink() ?>"
    rel="bookmark" title="Permanent Link to
    <?php the_title(); ?>"><?php the_title(); ?></a></h2>
</div>
```

刷新网页



日志的时间和分类都显示出来了，不过不够美观，接下去样式化

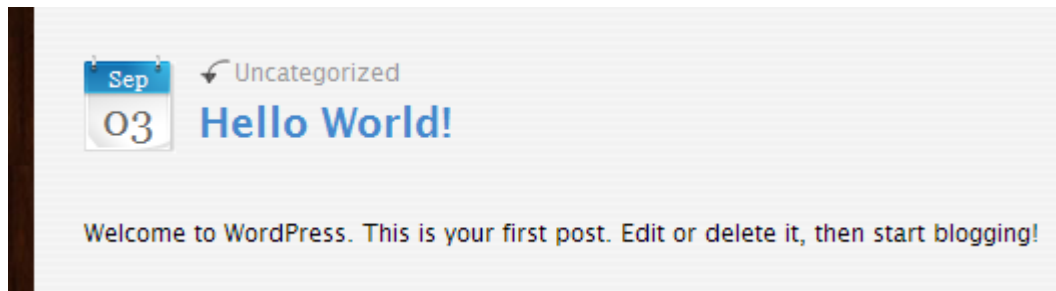
```
.post .date {
  height: 50px;
  width: 45px;
  background: url(images/calendar.gif) no-repeat;
  font: normal 22px Georgia,"Times New Roman",Times,serif;
  color: #666666;
  text-align: center;
  padding: 0px 2px 0 0;
  line-height: 100%;
  float: left;
}
.post .date span {
  height: 16px;
  display: block;
  font: normal 11px Georgia,"Times New Roman",Times,serif;
  color: #ffffff;
  text-align: center;
  padding-top: 5px;
}
```

这部分是设置发表时间，字体 Georgia 比较好看，特别是数字部分

```
.post .title {
  float: left;
  margin-left: 10px;
}
.category a{
  color: #999999;
}
.category a:hover {
  color: #000;
}
.category{
  background: url(images/cat-bg.gif) no-repeat;
  padding-left: 18px;
  margin-bottom: 10px;
}
```

整个 title 类别浮动到左边，这样就紧靠着左边的日期了，然后分类添加一张背景图片，

并设置左补白使文字不挡住背景图片，完成后看起来应该是这样的



对了，还要添加分页的链接，当日志多起来，一页显示不了的时候，就需要分页了，不过 wordpress 自带的分页功能比较简单，只有类似的上一页，下一页，而没有 1, 2, 3, 4 这样可以直接点击页数来转到对应页。如果你需要这个功能，那么有个叫做 wp-pagenavi 的插件可以帮到你，另外，撰写这个插件的作者 GamerZ 还撰写其他很多经典而热门的插件，比如 wp-polls 等，你可以关注一下。

我们这里使用的是 wp 原生的方法来显示分页

在`<?php endwhile; ?>`之后添加

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

这里有两个函数，一个是 `next_posts_link`，用来显示较旧的文章，而另一个是 `previous_posts_link`，用来显示较新的文章，他们都有一个字符串参数用来显示链接，你可以改成其他的文字。

日志部分到这里基本上已经完成了。

Searchform.php

这个文件很简单，里面只有这么几行

```
<form method="post" id="searchform"
  action="<?php bloginfo('url'); ?>/">
<input type="text" value="<?php the_search_query(); ?>"
name="s" id="s" autocomplete="off" />
<input type="submit" id="searchsubmit" value="Search" />
</form>
```

这只是一个简单的表单，里面有一个输入框，用来输入要搜索的内容，还有一个提交按钮

不过单单这个文件，并不会显示出一个搜索栏出来，要显示它，需要在其他地方，比如 sidebar 调用它，在讲到 sidebar 的时候再来说明，这里先放一下，只要明白，这个文件预先定义了一个搜索表单就可以了。

Functions.php

这里是一个让你自定义函数的地方，你可以把所有自己定义的函数写到这里

Sidebar.php

Sidebar 就是侧边栏，一般是用来显示一些辅助信息的，比如分类，存档，好友链接，搜索栏，广告栏等等，本主题很简洁，在侧边栏只显示了分类，存档，搜索

先大略样式化一下侧边栏


```
#sidebar{
    width:275px;
    float:right;
    color:#333;
    margin:35px 25px 0 0;
    display:inline;
}
#sidebar a{
    color:#333;
}
```

侧边栏宽度为 275px，右浮动，设置上边距 35px，右边距 25px，display:inline 上面已经解释过了

来看一下搜索

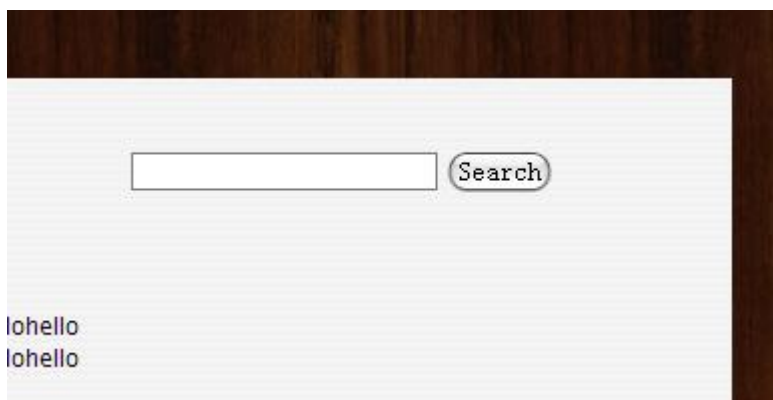
上面已经在 searchform.php 文件里定义了一个搜索框，接下去就要调用它了，调用方法很简单

在<div id="sidebar"></div>标签之间加入以下代码

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

include 表示要调用一个外部文件，TEMPLATEPATH 是一个常量，表示主题路径，点号用来串联两个字符串。

保存，刷新浏览器，就会在侧边栏出现一个搜索框，如图



默认的显示似乎还可以，不是很难看，不过这不是我想要的样式

```
#searchform{
  width:230px;
  height:62px;
  background:url(images/search-bg.gif) no-repeat left top;
  margin-bottom:30px;
}
```

设置搜索栏的宽度的高度，并设置了一张背景图片，并让它只显示图片的上半张

```
form#searchform.search-focus{
  width:230px;
  height:62px;
  background:url(images/search-bg.gif) no-repeat left bottom;
}
```

设置背景图片，不过这次让它显示下半张

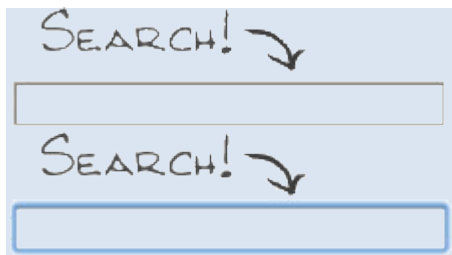
```
#searchform #s{
  width:208px;
  height:16px;
  background:none;
  border:none;
  margin:40px 0 0 5px;
}
```

设置输入框的位置的大小和位置

```
#searchsubmit{
  display:none;
}
```

隐藏搜索按钮

搜索栏的背景图片是这样的



除了搜索栏的边框部分不一样,其他部分都是一样的,平时显示的时候是上半张,当你点击它的时候就显示下半张图片,那么怎么样来判断何时点击呢?css 里有个:focus 伪类,不过 ie6 并不支持,另外,这个伪类是针对输入框的,所以这个方法并不适合,下面要实现的是,当点击输入框时,其父容器#searchform 就会产生一个类别叫做 search-focus,当失去焦点时,移除这个类别。怎么实现?只能靠 javascript 了。这里我使用了 jquery,在下面一章会比较详细的介绍一下 jQuery。

在<head></head>之间加入

```
<script type="text/javascript" src="<?php
bloginfo('template_url'); ?>/js/jquery-1.3.2.min.js"> </script>
<script type="text/javascript" src="<?php
bloginfo('template_url'); ?>/js/global.js"> </script>
```

这里载入两个 js 文件,一个就是 jquery 文件,一个是自定义的 js 文件,在 global 输入以下

代码

```
$(document).ready(function(){
    $('#s').focus(function(){
        $(this).parent('#searchform').addClass('search-focus');
        return false;
    });
    $('#s').blur(function(){
        $(this).parent('#searchform').removeClass('search-focus');
        return false;
    });
});
```

这段 js 代码的作用就是,当点击#s,也就是搜索框时,对其父容器#searchform 增加一个叫做 search-focus 的 class,而当失去焦点时,则移除这个 class

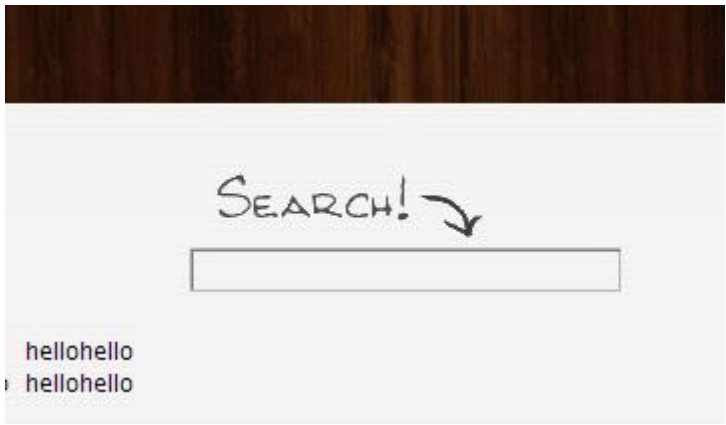
来看一下先前 css 的设置

```
form#searchform.search-focus{
    width:230px;
    height:62px;
    background:url(images/search-bg.gif) no-repeat left bottom;
}
```

这里不能将首行改成`.search-focus`，虽然两者指的是同一个东西，因为 `class` 的优先级没有 `id` 高，所以直接设置 `.search-focus` 并没有效果，故把 `id` 和 `class` 设置在一起。

设置好之后，来看一下实际的效果图

这是没有点击时候的图片



这是鼠标点击了搜索框之后的图片，和苹果系统非常的像



接下去要显示分类和存档

在`<?php include (TEMPLATEPATH.'/searchform.php'); ?>`这一行后加入

```
<div class="minibox1">
    <h3>category</h3>
    <ul> <?php wp_list_categories('title_li='); ?></ul>
</div>
<div class="minibox2">
    <h3>archive</h3>
    <ul><?php wp_get_archives('type=monthly'); ?> </ul>
</div>
<div class="clear"></div>
```

分别有两个小箱子，将分类和存档包裹起来，然后用一个空的箱子来清除所有的浮动，这个

前面也已经讨论过为什么要清除浮动

来设置 css 吧

```
#sidebar h3{
    font-size:16px;
    color:#99cc00;
    text-transform:capitalize;
    margin-bottom:7px;
    font-weight:normal;
}
```

标题的颜色，大小，首字母大写

```
.minibox1{
    width:125px;
    float:left;
}
```

```
.minibox2{
    width:125px;
    float:left;
    margin-left:25px;
    display:inline;
}
```

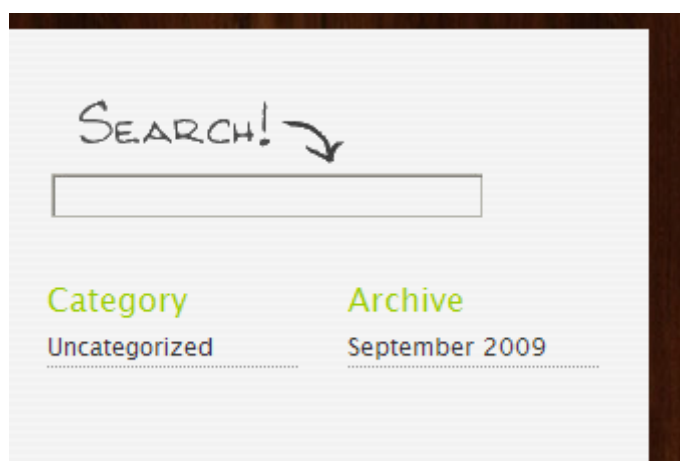
设置两个箱子的宽度，并让它们浮动到左边

```
.minibox1 li{  
  border-bottom:1px dotted #999;  
  padding-bottom:2px;  
  margin-top:3px;  
}
```

```
.minibox2 li{  
  border-bottom:1px dotted #999;  
  padding-bottom:2px;  
  margin-top:3px;  
}
```

设置一下列表元素的样式，每个列表元素下面会有点线

效果图如下



这样，侧边栏就设置好了

Single.php

这个是用来显示单篇日志的布局的，copy 一份 index.php，然后把它改名成 single.php.

单篇日志和首页最主要的不同之处是日志后面可以显示评论和评论框

所以在`<?php endwhile; ?>`这一行之前加入下面这句代码

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

而评论部分怎么显示，则是在 comments.php 里设置的。

另外，想要在文章的最后显示 tag，加入下面一句

```
<?php the_tags( '<p>Tags: ', ' ', ' ', '</p>' ); ?>
```

其中有三个参数，第一个参数是前缀，中间的参数是标签之间分隔的字符，这里设置的是逗号，最后一个是后缀，最后在浏览器里显示出来的源码看上去是这样

```
<p>Tags: foo, bar</p>
```

Page.php

这个文件是用来显示页面(page)的布局的，同理，直接拷贝 index.php，改名成 page.php

就可以了，因为是页面文件，所以我不需要让它显示日期，分类，直接删除相关的代码即可，

如果不想页面有评论，也可以像我一样不在页面加入

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

Comments.php

这个文件是用来显示评论和评论框的，前面在 single.php 已经加入

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

即使没有 comments.php 这个文件，还是可以显示和提交评论的，不过 comments.php

文件可以让我们自定义想要显示的内容

```
<?php // Do not delete these lines
    if ('comments.php' == basename($_SERVER['SCRIPT_FILENAME']))
        die ('Please do not load this page directly. Thanks!');
    if (!empty($post->post_password)) { // if there's a password
        if ($_COOKIE['wp-postpass_' . COOKIEHASH] !=
$post->post_password) { // and it doesn't match the cookie
            ?>
            <p class="nocomments">This post is password protected.
Enter the password to view comments.</p>
            <?php
                return;
            }
        }
        /* This variable is for alternating comment background */
        $oddcomment = 'alt';
    ?>
```

这部分不要改动，直接原样照抄放在 comments.php 文件的最前面即可

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

判断有没有评论，如果有的话，执行下面的代码

```
<div class="navigation">
    <div class="alignleft"><?php previous_comments_link() ?>
    </div>
    <div class="alignright"><?php next_comments_link() ?></div>
</div>
```

评论分页时的导航内容


```
<ol class="commentlist">
```

定义了一个有序的列表，并取名叫 `commentlist`，每条评论就是这个有序列表的列表元素

```
<?php $i=1; foreach ($comments as $comment) : ?>
```

首先设置一个计数器 `i`，然后通过 `foreach` 来遍历所有的评论

```
<li id="comment-<?php comment_ID() ?>" <?php if  
($comment->comment_author_email == get_the_author_email()) echo  
'class="admin"'; ?> >
```

判断评论者的邮箱是不是和作者的邮箱符合，如果符合，就添加一个叫做 `admin` 的类别，

在 `css` 样式化，可以把作者的评论和一般游客的评论醒目的区别开来

```
<div class="comment-meta"><?php echo"<em>$i.</em>&nbsp;" ;  
comment_date('M jS, Y') ?> at <?php comment_time() ?>
```

通过先前设置的计数器显示是第几个评论，同时显示发表评论的时间

```
<span class="author"><?php comment_author_link() ?> says:</span>  
<span></span></div>
```

显示评论者的名字

```
<div class="comment-entry">  
    <div class="avatar-place"><?php echo get_avatar( $comment,  
32 ); ?></div>
```

显示头像

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

显示评论正文

```
<div class="clear"></div>
```

清除浮动（因为头像设置了浮动）

```
</div>
```

```
</li>
```

```
<?php $i++; endforeach; ?>
```

```
</ol>
```

评论显示结束

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

Else 部分显示的是没有评论的时候

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

```
<?php echo "<p>no comment yet.</p>" ;?>
```

显示没有评论

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

```
<?php echo "<p>comments are closed.</p>" ; ?></p>
```

显示评论关闭

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

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

结束判断

接下来添加评论框

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

判断评论有没有开启，开启的话才需要加上一个评论框，关闭的话就不用添加了。

```
<h3 id="respond">Leave a reply</h3>
```

设置一个小标题

```
<?php if ( get_option('comment_registration')
```

```
&& !is_user_logged_in() ) : ?>
```

```
<p><?php printf(__('You must be <a href="%s">logged in</a> to post  
a comment.'), wp_login_url( get_permalink() ));
```

```
?></p>
```

如果设置了注册用户才能发表评论，就显示一句话来提示你登录，才能发表评论

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

否则，开启了游客评论

```
<form action="<?php echo
```

```
get_option('siteurl'); ?>/wp-comments-post.php" method="post "  
id="commentform">
```

这是表单，设置 action 的地址，还有方法设置成 post，而不是 get

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

```
<p><?php printf(__('Logged in as <a href="%1$s">%2$s</a>.'),  
get_option('siteurl') . '/wp-admin/profile.php',
```

```
$user_identity); ?> <a href="<?php echo
```

```
wp_logout_url(get_permalink()); ?>" title="<?php _e('Log out of this
account'); ?>"
<?php _e('Log out &raquo;'); ?></a></p>
```

如果用户登录了，就显示已经登录，并显示一个登出的链接

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

如果没有登录

```
<p><input type="text" name="author" id="author" value="<?php echo
$comment_author; ?>" size="22" tabindex="1" /></p>
```

```
<p><input type="text" name="email" id="email" value="<?php echo
$comment_author_email; ?>" size="22" tabindex="2" /></p>
```

```
<p><input type="text" name="url" id="url" value="<?php echo
$comment_author_url; ?>" size="22" tabindex="3" /></p>
<?php endif; ?>
```

用来输入姓名，邮箱和个人网页

```
<p><textarea name="comment" id="comment" cols="100%" rows="10"
tabindex="4"></textarea></p>
```

这是显示评论的输入框

```
<p><input name="submit" type="submit" id="submit" tabindex="5"
value="<?php _e('Submit Comment'); ?>" />
<?php comment_id_fields(); ?>
</p>
```

提交按钮

```
<?php do_action('comment_form', $post->ID); ?>
```

```
</form>
```

```
<?php endif; // If registration required and not logged in ?>
```

```
<?php endif; // if you delete this the sky will fall on your head ?>
```

一个完整的评论已经完成，现在要样式化了。

```
.commentlist{  
  padding:20px 0;  
}
```

这是整个评论列表，设置上下的留白

```
.commentlist li{  
  margin:0 0 20px 0;  
}
```

每条评论之间增加一些空隙

```
.comment-meta{  
  background:#D1D6E0;  
  height:25px;  
  line-height:25px;  
  color:#333;  
  padding-left:10px;  
}
```

这里显示的评论的作者，发表时间，line-height 是让文字上下居中的好方法

```
.comment-meta a{  
  color:#333;  
}
```

链接颜色

```
.admin .comment-meta{  
  background:#598724;  
  color:#fff;  
}
```

这里设置博主留言的样式，方便区别是一般用户的评论还是博主自己的评论

```
.admin .comment-meta a{  
  color:#fff;  
}
```

链接颜色

```
.author{  
  font-weight:bold;  
  margin-left:20px;  
}
```

```
.comment-entry{  
  padding:10px 0 0 0;  
}
```

```
.comment-entry p{  
  margin:0;  
}
```

这里评论正文部分的设置，主要设置一下留白和边界

```
.avatar-place{  
  float:right;  
  margin:0 0 7px 15px;  
  padding:0;  
}
```

这是头像，让它显示到最右边

```
#author,#email,#url{  
  border:2px solid #b3b2b2;  
  color:#000;  
  width:200px;  
  padding:3px;  
  font-size:14px;  
  font-weight:bold;  
}
```

```
#comment{  
  border:2px solid #aaa;  
  color:#000;  
  width:400px;  
  height:160px;  
  font-size:14px;  
  padding:3px;  
  background:none;  
}
```

评论框部分的样式

```
#submit{  
  width:135px;  
  height:25px;  
  background:url(images/submit-btn.gif) no-repeat left bottom;  
  border:none;  
  font-weight:bold;  
  margin-top:20px;  
}
```

提交按钮的设置，我设置了一个苹果风格的按钮来替换系统难看的按钮

```
#submit:hover{
  width:135px;
  height:25px;
  background:url(images/submit-btn.gif) no-repeat left top;
  border:none;
}
```

当鼠标移动到按钮上，更改按钮样式

```
#commentform{
  width:420px;
  padding:20px 0 40px 0;
}
```

```
#author:focus,#email:focus,#url:focus,#comment:focus{
  border:2px solid #4488cc;
}
```

当鼠标激活输入框的样式，不过 ie6 并不支持:focus


```
#author{
  padding-left:20px;
  background:url(images/author.gif) no-repeat 0 50%;
}
#email{
  padding-left:20px;
  background:url(images/email.gif) no-repeat 0 50%;
}


#url{
  padding-left:20px;
  background:url(images/url.gif) no-repeat 0 50%;
}
```


6. Sep 9th, 2009 at 11:22 am
jimmy says:

测试一下评论

Leave a reply


jimmy


frozenthroner88@gmail.com



Submit Comment

因为编辑本文档时处于离线状态，所以没有显示头像。

这样评论部分也设置好了。

Footer.php

Footer 部分我设置的主要是用来显示受欢迎的文章，最新评论和站点介绍，所以需要三个箱子把这三个部分包裹起来，首先设置第一个箱子，用来显示热门文章

```
<div class="popular-posts">
```

新建箱子

```
<h3>Popular Posts</h3>
```

设置小标题

```
<ul>
```

开始列表

```
<?php $result = $wpdb->get_results("SELECT
comment_count,ID,post_title FROM $wpdb->posts ORDER BY comment_count
DESC LIMIT 0 , 6");
```

对数据库进行操作，数据库里选择 comment_count,ID,post_title 这三个部分，按照 comment_count 排序，然后限制只提取最前面的六条，你可以修改 6 这个数字，改成你

想显示的条数

```
foreach ($result as $post) {
setup_postdata($post);
$postid = $post->ID;
$title = $post->post_title;
$commentcount = $post->comment_count;
if ($commentcount != 0) { ?>
<li><a href="<?php echo get_permalink($postid); ?>" title="<?php echo
$title ?>">
<?php echo $title ?></a> {<?php echo $commentcount ?>}</li>
```

显示文章

```
<?php } } ?>
</ul>
</div>
```

接下去的部分显示最新评论

```
<div class="recent-comments">
<h3>Recent Comments</h3>
<ul>
<?php
global $wpdb;
$sql = "SELECT DISTINCT ID, post_title, post_password, comment_ID,
comment_post_ID, comment_author, comment_date_gmt,
comment_approved,comment_type,comment_author_url,
```



```
SUBSTRING(comment_content,1,36) AS com_excerpt
```

你可以将 36 改成你想显示每条评论的字数

```
FROM $wpdb->comments
LEFT OUTER JOIN $wpdb->posts ON ($wpdb->comments.comment_post_ID =
$wpdb->posts.ID)
WHERE comment_approved = '1' AND comment_type = '' AND
post_password = ''
ORDER BY comment_date_gmt DESC
LIMIT 4";
```

你可以把 4 改成你想显示的评论条数

```
$comments = $wpdb->get_results($sql);
$output = $pre_HTML;
foreach ($comments as $comment) {
$output .= "<li>".strip_tags($comment->comment_author).": ". "<a
href=\"\" .
get_permalink($comment->ID).
\"#comment-\" . $comment->comment_ID . "\" title=\"on ".
$comment->post_title . "\"><br />\" .
strip_tags($comment->com_excerpt)
.\"</a>\".\"</li>\";}
$output .= $post_HTML;echo $output;?>
</ul>
</div>
```

这里是输入的显示格式，你可以自行修改，不如你不了解 php 的，改起来可能有点难度

```
<div class="page-info">
<h3>About</h3>
<p>ArtLess is all about wordpress, emacs, web design, and web2.0. It
is designed and maintained by Jimmy NG.</p>
<p>If you have any questions or suggestions,please <a href="<?php echo
get_option('home'); ?>/contact/">let me know</a>,thank you.</p>
</div>
```

这里是关于个人和网站的介绍

```
<div class="copyright">
&copy;<a href="<?php bloginfo('url'); ?>"><?php
bloginfo('name'); ?></a> <?php echo date("Y"); ?>
. Powered by <a href="http://wordpress.org">WordPress</a>. All rights
```

```
reserved.
```

```
</div>
```

这里是一些版权声明等

最后添加

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

这是一个 hook

```
<div class="clear"></div>
```

清除浮动

接下去就要样式化这三个箱子了，这部分 css 比较简单，不一一说明了。

```
#footer{  
    width:840px;  
    background:#000;  
    color:#fff;  
    clear:both;  
    margin:0;  
    padding:15px 0;  
}
```

```
.popular-posts{  
    width:200px;  
    float:left;  
    margin-left:15px;  
    color:#555;  
    display:inline;  
}
```

```
.popular-posts h3{  
    color:#779d03;  
    font-size:16px;  
    font-weight:normal;  
    border-bottom:1px dotted #333;  
    padding-bottom:5px;  
}
```

```
.popular-posts ul li{  
  border-bottom:1px dotted #333;  
  padding:3px;  
}
```

```
.recent-comments{  
  width:200px;  
  float:left;  
  margin-left:15px;  
  color:#555;  
  display:inline;  
}
```

```
.recent-comments h3{  
  color:#779d03;  
  font-size:16px;  
  font-weight:normal;  
  border-bottom:1px dotted #333;  
  padding-bottom:5px;  
}
```

```
.recent-comments ul li{  
  border-bottom:1px dotted #333;  
  padding:3px;  
}
```

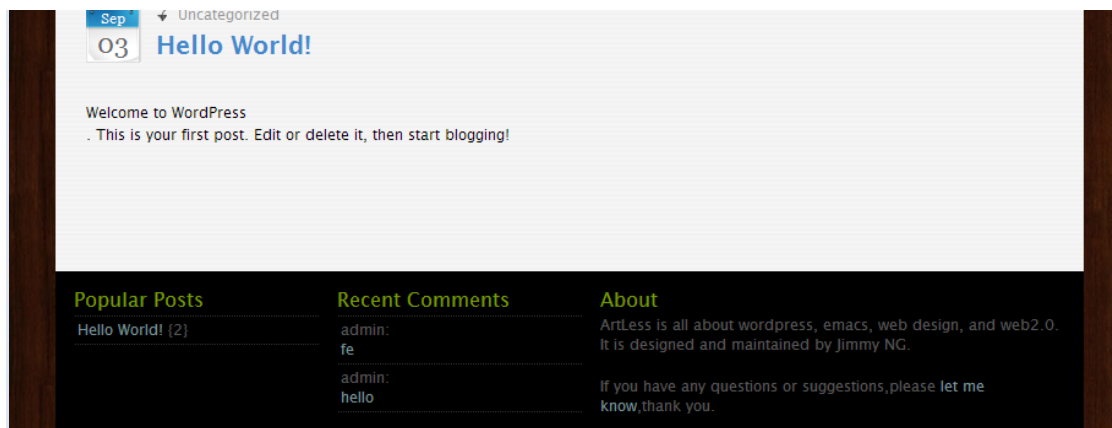
```
.page-info{  
  width:380px;  
  float:left;  
  margin-left:15px;  
  color:#555;  
  display:inline;  
}
```

```
.page-info h3{  
  color:#779d03;  
  font-size:16px;  
  font-weight:normal;  
  padding-bottom:5px;  
}
```

```
#footer a{
```

```
color:#759398;
}
```

最后看起来 footer 部分的样子



404.php

404 是什么？404 是 http 的出错代码，代表文件没有找到，当你在网页上输入了一个不正确的网址，或许你的文章已经删除了，那么服务器找不到这个页面，就会返回 404 错误，这个 404.php 就是当服务器找不到页面时，应该怎么显示。这个文件比较简单，不需要像首页，日志那样需要一个 loop，

```
<?php get_header(); ?>
<div id="content">
  <div class="post">
    <h2>Error 404</h2>
    <div class="entry">
      <p>error 404,not found</p>
    </div>
  </div>
</div>
<?php get_sidebar(); ?>
<?php get_footer(); ?>
```

那么当出现 404 错误的时候，就只会显示 error 404,not found 这么一句话了

screenshot.png

每个主题的文件夹里都会有一张叫做 screenshot.png 的图片，这张图片就是你在 wordpress 后台浏览主题时的预览图，其实并没有一定的大小限制，不过惯例是 300px 乘 225px。当然没有这个文件也行，你的主题并不会出错，只是在后台预览的时候就一片空白了。下面来简短的总结一下：

一个主题文件应该包括

style.CSS	样式表文件
index.php	主页文件
single.php	日志文件
page.php	页面文件
archvie.php	分类和日期存档页文件
searchform.php	搜索表单文件
search.php	搜索页面文件
comments.php	评论区域文件
404.php	404 错误页面
header.php	网页头部文件
sidebar.php	侧边栏文件
footer.php	网页底部文件
screenshot.png	后台缩略图显示

一路走来，我们已经比较详细的介绍了上面的大部分文件，因为有些文件不是必须的，而且对我这个主题来说不重要，就省略了。

第四章：进阶篇

设置模板

Wordpress 除了 post，就是我们所说的日志，还有 page，就是页面，页面一般用来放一些比较长久性的东西，比如个人介绍啊，下载专区啊，邮件联系啊等等，你可以直接在 wp 的后台写页面，也可以使用模板，然后在后台撰写页面的时候套用它。那么模板有什么好处呢？没有模板不是一样可以在后台撰写页面吗？

模板的最大好处就是可以运行 php 代码

如果你想用一个单独的页面来显示好友链接，你可能会在页面写入

```
<ul>
<?php get_links('-1', '<li>', '</li>', '', FALSE, 'id', FALSE, FALSE,
-1, FALSE); ?>
</ul>
```

不过让你失望的是，页面并不会显示出一个好友列表来，而是会照原样输出，因为默认情况下，wordpress 并不会执行页面里的 php 代码。因为<>之类的特殊字符都被翻译成<,\$gt;了。这是你可能会耍点小聪明，我不在 visual 模式下编辑，我在 html 源码下编辑，是不是可以显示了呢？答案当然也是否定的。

你在源码下输入以上代码，输入的结果看起来可能会是这样的

```
', ' ', ", FALSE, 'id', FALSE, FALSE,
-1, FALSE); ?>
```

这个时候，你只要撰写一个简单的页面模板，就可以来显示好友链接了。首先在主题目录下新建一个叫 links.php 的文件，在开始处加入下面几行代码，来表示这是一个模板文件。

```
<?php
/*
Template Name: links
*/
```

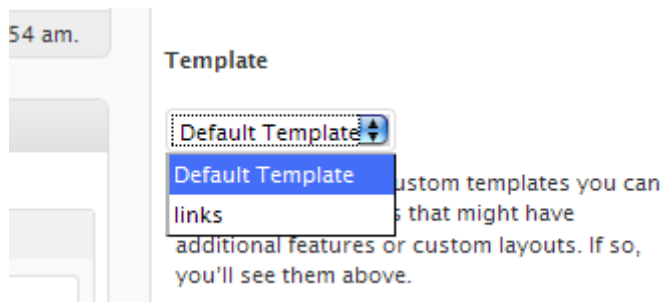
```
?>
```

模板文件不需要 loop , 因为只是单独的一篇而已 , 只要简单的修改一下 page.php 就可以了

```
<?php get_header(); ?>
<div id="content">
    <div class="post">
        <h2>
            <?php the_title();?>
        </h2>
        <div class="entry">
            <ul>
                <?php get_links('-1', '<li>', '</li>', '', FALSE, 'id',
FALSE, FALSE, -1, FALSE); ?>
            </ul>
        </div>
    </div>
</div>
<?php get_sidebar(); ?>
<?php get_footer();?>
```

这样 , 一个模板文件就撰写好了 , 怎么使用呢 ?

进入后台 , 点击撰写新页面 , 输入标题 , 正文部分留空 , 然后在右下方找到模板。选择一个模板 , 比如这里是 links , 然后发布就可以了



使用 jQuery

jQuery 是目前相当流行的一个 js 库,由 John Resig 建立。它的理念是“写的少做的多”。

无论你具有什么样的程序设计背景,都可以用它来加强网站的功能,简单易学,具有优秀的跨平台性。它的使用方法也很简单,首先从 jquery.com 下载源码,然后放到某个文件夹,

比如 sample/js/里,然后在<head></head>标签之间加入以下代码

```
<script type="text/javascript" src="
<?php bloginfo('template_url'); ?>/js/jquery-1.3.2.min.js">
</script>
```

这样就可以使用 jquery 了,为了方便,把所有的自定义代码放到单独的一个文件中,比如叫做 global.js

同理,在<head></head>之间加入

```
<script type="text/javascript" src="
<?php bloginfo('template_url'); ?>/js/global.js"> </script>
```

将下列代码写入 global.js 中

```
$(document).ready(function(){
    //所有代码将会放到这里
});
```

只要等到浏览器将 dom 树载入完成后,就可以使用内部定义的代码了,而不必等到所有的内容都载入完成(比如有大量图片的时候,完全载入将会是一个漫长的过程)。

jQuery 有强大的选择器,支持类似 css 的语法,比如\$('p')就是全体 p 元素的集合, \$('post ul')则是类别 post 下所有 ul 的集合,这些看上去似乎很稀松平常,和普通的 css 选择器差不多嘛,但是 jQuery 的选择器并不局限于此,我们会在接下来的实例中看到比较复杂的选择器

接下来看一下实际的应用

样式化下载链接

设想一下这样的情景，你在博客里提供了一个下载链接，你一定希望你的下载链接是十分醒目的，别人一看就知道，而不是和其他普通的链接混杂在一起，这时你会怎么样做？或许

你已经想到了，每次增加下载链接的时候，指定给它一个图片，比如代码或许是这样的

```
<a href="xxx.zip"></a>
```

这样就显示了一张 download.gif 图像，点击它时下载 xxx.zip 文件

这似乎已经解决了问题，但是每次增加下载项的时候，你都需要手工指定图片，这会是一个枯燥而烦人的工作，有更好的解决方法吗？有，让 jquery 帮到你

在 global.js 写入

```
$( 'a[href$=zip]' ).addClass( "download" );  
$( 'a[href$=rar]' ).addClass( "download" );  
$( 'a[href$=gz]' ).addClass( "download" );  
$( 'a[href$=7z]' ).addClass( "download" );
```

简单来说，上面四行代码会检查所有 a 标签里的 href 属性，如果以 zip,rar,gz 或 7z 结尾，

则自动给 a 标签加上一个 download 的类别

原始的代码

```
<a href="foo.rar">foo</a>
```

会自动变成

```
<a href="foo.rar" class="download">foo</a>
```

接下来在 style.css 里加入

```
a.download {  
    display: block;  
    width: 235px;  
    height: 37px;  
    background: url(images/download.gif) no-repeat;  
    text-indent: -9999px;  
}
```

效果看起来是这样的



在新窗口打开外部链接

当游客好不容易访问到了你的网站,你一定不希望你的游客在点击了一个链接之后,直接在原窗口打开新链接,你可能希望是在新的 tab 打开链接,这是有 a 标签里的 target 属性控制的,当 target 属性设置成_blank 时,则会在新窗口打开链接,所以需要对外部链接设置这个属性为_blank,同理,你不会希望是自己手工一个一个添加,让我们看看 jquery 怎么减轻您的负担。

```
$( "a[href^=http://]").not( "[href*=foo.com]" )  
.attr( "target", "_blank" );
```

首先限定所有 href 属性以 http://开头的 a 标签,然后过滤掉 href 里包含 foo.com 的 a 标签(foo.com 替换成你自己博客地址),最后添加属性 target 为_blank

显示与隐藏

这是一个比较平常的需求,你点击一个标题,会隐藏其下的列表,再次点击该标题,隐藏的列表又会重新显示出来,看看是怎么做的

Html 代码如下

```
<h3 class="foo">foo</h3>  
<ul class="bar">
```

```
<li>bar</li>
<li>bar</li>
<li>bar</li>
<li>bar</li>
</ul>
```

Js 代码如下

```
$('.foo').click(function(){
    $('.bar').toggle();
});
```

首先选择类别 foo，对点击事件添加一个匿名函数，然后对类别 bar 作出一些动作，toggle 方法会在隐藏和显示之间切换，就是说，原本是显示的，执行完 toggle 后会隐藏起来，如果原本是隐藏的，则执行完 toggle 之后会显示出来。

如果一开始想要列表隐藏，点击标题后让它显示出来，再次点击又隐藏列表，该怎么做？

只要多加一行就可以了

```
$('.bar').hide();
$('.foo').click(function(){
    $('.bar').toggle();
});
```

hide()方法可以让选定的元素集合隐藏起来。

表单验证

表单验证是一个比较重要的环节，这里通过 js 来进行简单的本地验证，如果有错，直接提示用户，避免用户按了提交按钮后才出现错误的提示。注意，本地验证并不能代替服务端的验证

以下是 comments.php 里关于评论者名字和邮箱的代码部分

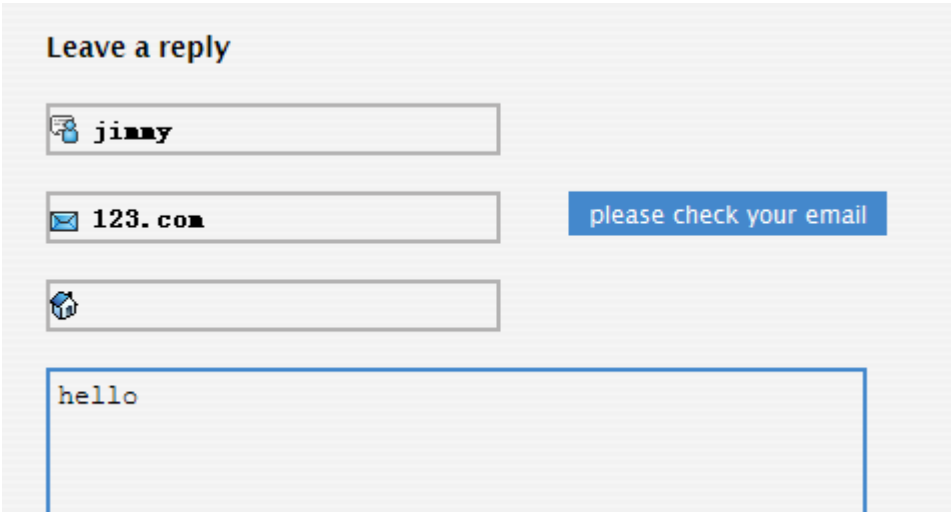
```
<p><input type="text" name="author" id="author" value="php echo
$comment_author; ?"
size="22" tabindex="1" /><span class="check1">please check your
```

```
name</span></p>

<p><input type="text" name="email" id="email" value="<?php echo
$comment_author_email;
?>" size="22" tabindex="2" /><span class="check2">please check your
email</span></p>
```

在输入框后添加了 span，默认是隐藏的，如果输入了一个空白的名字或者一个格式不正确的 email，在焦点离开输入框的时候显示出来

效果图如下



代码如下

```
$( '.check1,.check2' ).hide();
    $("#author").change(function(){
        if($.trim($(this).val())=='')
            $(".check1").show(1000);
        else
            $(".check1").hide(1000);
    });
    $("#email").change(function(){
var re=
/^[a-zA-Z0-9_\. \-]+\@(([a-zA-Z0-9 \-]+\.)+([a-zA-Z0-9]{2,4})+)$/;
        if(!re.test($(this).val()))
            $(".check2").show(1000);
        else
            $(".check2").hide(1000);
    });
```

首先将.check1,.check2 隐藏起来，然后选择#author，在有改变（如失去焦点时）执行函

数,首先判断输入的值在去除首位空白后是不是空白字符,如果是,则显示 check1 的内容,否则,就隐藏 check1。

同理,选择#email,在有改变时执行函数,判断输入的 email 地址是不是满足定义的格式(一个正则表达式),如果不满足,则显示 check2 的内容,否则,隐藏 check2。

显示相关日志

根据日志的标签来显示相关的日子,必须放在 loop 里

```
<?php
$tags = wp_get_post_tags($post->ID);
if ($tags) {
    echo 'Related Posts';
    $first_tag = $tags[0]->term_id;
    $args=array(
        'tag__in' => array($first_tag),
        'post__not_in' => array($post->ID),
        'showposts'=>5,
        'caller_get_posts'=>1
    );
    $my_query = new WP_Query($args);
    if( $my_query->have_posts() ) {
        while ($my_query->have_posts()) : $my_query->the_post(); ?>
            <p><a href="<?php the_permalink() ?>" rel="bookmark"
title="Permanent Link to <?php the_title_attribute(); ?>"><?php
the_title(); ?></a></p>
            <?php
        endwhile;
    }
}
?>
```

显示最新文章

如果你想在 sidebar 或者其他地方显示你最新发表的文章，你可以加入这么一句话

```
<?php get_archives('postbypost',5); ?>
```

其中 5 是你想要显示的文章标题的数目

显示特定分类的文章

如果你的页面布局比较另类，比如将文章按分类集中到一起显示，而不是按照时间顺序将所

有的日子排列到一起，那么你该试试下面的方法，这段代码列出分类 1 中的最新 5 篇文章

```
<?php query_posts('cat=1&showposts=5'); ?>
    <?php if (have_posts()) : ?>
        <?php while (have_posts()) : the_post(); ?>
            <h2><a href="<?php the_permalink() ?>" rel="bookmark" >
<?php the_title(); ?></a></h2>
            <?php endwhile; ?>
        <?php endif; ?>
```

随机顺序显示日志

将下面一行代码放到 loop 前，就可以以随机顺序来显示日志了

```
<?php query_posts('orderby=rand'); ?>
```

不同的节日显示不同的消息

将下面代码放到你想显示的位置，就会在不同的节日那天显示不同的问候语

```
<?php
if ((date('m') == 12) && (date('d') == 25))
    echo "<p>圣诞节快乐</p>";
if ((date('m') == 1) && (date('d') == 1))
```

```
echo "<p>元旦快乐</p>";  
if ((date('m') == 2) && (date('d') == 14))  
    echo "<p>情人节快乐</p>";  
?>
```

一些 CSS 的小技巧

用一张图片来代替两张图片

比如当鼠标移动到一个元素上时，更改背景图片，或许你会这么做

```
#foo{  
    background:url(image1.gif) no-repeat;  
    width:100px;  
    height:100px;  
}  
#foo:hover{  
    background:url(image2.gif) no-repeat;  
}
```

这会有什么问题呢？因为是两张图片，所以当页面第一次载入完成的时候，只有 image1.gif 载入，而鼠标移动到 #foo 元素上时，浏览器才会重新发送请求，从服务器取得 image2，因为网速的问题，图片传输过来需要时间，虽然小图片可能用不了 1 秒钟，但是从视觉上看，还是会产生闪烁感。解决这个问题，就是用一张图片来替代

举个例子，假设原始的 image1 是 100px 见方的银色苹果



而 image2 是 100px 见方的蓝色苹果



这么我们制作这么一张图片，宽度是 100px，高度是 200px（实际没有粉色背景，为了便于读者区分实际图片和白色的文档背景）



当文档载入时显示的是上半张图片，而当鼠标移动到#foo 上是显示下半张图片，来看看 css

怎么写

```
#foo{  
  background:url(image3.gif) no-repeat left top;  
  width:100px;  
  height:100px;  
}  
#foo:hover{  
  background:url(image3.gif) no-repeat left bottom;  
}
```

通过设置背景图片的大小和位置（注意 top 和 bottom），因为是一张图片，所以切换时完全没有延时，问题解决了。

本主题的 search 栏部分就是用了这个技巧，不过用这个技巧有一点很重要，就是上下两个

图片位置要对齐，不然上下切换的时候位置会错位，我们看到的不单单是颜色的改变了，苹果也会动起来。怎么让它位置对齐？这是 photoshop 的技巧，这里省略先。

用背景图片替代链接图片

这在制作主页 logo 链接的时候会用到，要在头部显示一个个性化的 logo，然后连接到首页，很多人会这么做

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

这样就显示了一个 logo，而且是链接到某个网页的，这有什么不好吗？对搜索爬虫不好，搜索爬虫只认得文字，不认得图片，它知道这里有张图片，链接到了某个网页，另外屏幕阅读机也不认得图片。一个好的做法是

```
<a href="#" id="logo">blog title</a>
```

```
#logo{
  display:block;
  width:200px;
  height:40px;
  background:url(images/logo.gif) no-repeat;
  text-indent:-9999em;
}
```

通过设置 display 为 block，把内联元素变成块状元素显示，就可以指定其大小了，设置一个背景图片给它，然后把文字隐藏起来，这样显示效果和先前的一样，不过对搜索爬虫和屏幕阅读机来说，它们看到这是你的 blog title，链接到了首页，具有很清晰的文档结构。

另外你的 logo 不是单张图片，而是在大的背景图片里，也没有关系，只要指定链接的位置在你背景图片的 logo 部分上就好了，而不用制作一个透明的 gif 文件，然后做成图片链接放到背景 logo 的位置。

善用空的 span

span 你一定不会陌生，它和 div 差不多，不过它是内联的，而不像 div 是块状的，比如你要对一串文字的部分样式化，可以这么做

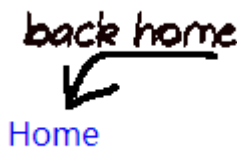
```
<p>hello,<span class="foo">nice</span> to meet you</p>
```

这样你就可以样式化 nice 这个单词了

那么一个空的 span 有什么用呢？它什么也没有啊？它太有用了，很多 amazing 的技巧其实就是用了空的 span 而已。让我们来看一个例子

[Home](#)

这是一个普通的链接，当鼠标移动到文字上时，在它的上面显示一张图片



Html 代码是这样的

```
<ul>
  <li> <a href="#" class="home">Home<span /></a></li>
</ul>
```

Css 部分

```
.home{
  position:relative;
}
.home:hover span{
  display:block;
  width:118px;
  height:50px;
  position:absolute;
```

```
top:-50px;  
left:0;  
background:url(home.gif) no-repeat;  
}
```

首先.home 设置相对定位，这样其子元素设置成绝对定位的时候，会根据它的位置来决定自身的位置，平时，span 是空的，什么都不会显示，当鼠标移动到文字上时，设置 span，让它具有一个背景图片，然后绝对定位它，让它定位到文字的上方，这样，图片就显示出来了。

附录 A: 常用函数*

Wordpress Header 头部 PHP 代码

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

网站标题

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

日志或页面标题

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

Wordpress 主题样式表文件 style.css 的相对地址

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

Wordpress 博客的 Pingback 地址

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

Wordpress 主题文件的相对地址

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

博客的 Wordpress 版本

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

Wordpress 博客的 Atom 地址

```
<?php bloginfo('rss2_url'); ?>
```

Wordpress 博客的 RSS2 地址

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

Wordpress 博客的绝对地址

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

Wordpress 博客的名称

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

网站的 HTML 版本

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

网站的字符编码格式

Wordpress 主体模板 PHP 代码

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

日志内容

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

确认是否有日志

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

如果有，则显示全部日志

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

结束 PHP 函数“ while”

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

结束 PHP 函数“ if”

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

header.php 文件的内容

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

sidebar.php 文件的内容

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

footer.php 文件的内容

```
<?php the_time('m-d-y') ?>
```

显示格式为“ 02-19-08”的日期

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

显示一篇日志的留言链接

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

显示一篇日志或页面的标题

```
<?php the_permalink() ?>
```

显示一篇日志或页面的永久链接/URL 地址

```
<?php the_category(' , ') ?>
```

显示一篇日志或页面的所属分类

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

显示一篇日志或页面的作者

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

显示一篇日志或页面的 ID

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

显示一篇日志或页面的编辑链接

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

显示 Blogroll 中的链接

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

comments.php 文件的内容

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

显示一份博客的页面列表

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

显示一份博客的分类列表

```
<?php next_post_link(' %link ') ?>
```

下一篇日志的 URL 地址

```
<?php previous_post_link('%link') ?>
```

上一篇日志的 URL 地址

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

调用日历

```
<?php wp_get_archives() ?>
```

显示一份博客的日期存档列表

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

显示较新日志链接(上一页)和较旧日志链接(下一页)

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

显示博客的描述信息

其它的一些 Wordpress 模板代码

```
/%postname%/
```

显示博客的自定义永久链接

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

搜索表单的值

```
<?php _e('Message'); ?>
```

打印输出信息

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

显示注册链接

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

显示登入/登出链接

```
<!--next page-->
```

在日志或页面中插入分页

```
<!--more-->
```

截断日志

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

显示管理员的相关控制信息

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

显示载入页面的时间

```
<?php echo get_num_queries(); ?>
```

显示载入页面查询

*注：以上内容转自帕兰映像翻译的 WordPress-Help-Sheet