

13-几个改进

1、在模型中指定排序

为了让文章（Post）按发布时间逆序排列，即最新发表的文章排在文章列表的最前面，我们对返回的文章列表进行了排序，即各个视图函数中都有类似于 `Post.objects.all().order_by('-created_time')` 这样的代码，这导致了很多重复。因为只要是返回的文章列表，基本都是逆序排列的，因此我们可以在 Post 模型中指定 Post 的自然排序方式。

Django 允许我们在 `models.Model` 的子类里定义一个 Meta 的内部类，这个内部类通过指定一些属性来规定这个类该有的一些特性，例如在这里我们要指定 Post 的排序方式。首先看到 Post 的代码：

【blog/models.py】

```
@python_2_unicode_compatible
class Post(models.Model):
    ...
    created_time = models.DateTimeField()
    ...

    def __str__(self):
        ...

    def get_absolute_url(self):
        ...
```

现在在 Post 类的内部定义一个 Meta 类，并指定排序属性：

【blog/models.py】

```
@python_2_unicode_compatible
class Post(models.Model):
    ...
    created_time = models.DateTimeField()
    ...

    def __str__(self):
        ...

    def get_absolute_url(self):
        ...

    class Meta:
        ordering = ['-created_time']
```

ordering 属性用来指定文章排序方式，['-created_time'] 指定了依据哪个属性的值进行排序，这里指定为按照文章发布时间排序，且负号表示逆序排列。列表中可以用多个项，比如 ordering = ['-created_time', 'title']，那么首先依据 created_time 排序，如果 created_time 相同，则再依据 title 排序。这样指定以后所有返回的文章列表都会自动按照 Meta 中指定的顺序排序，因此可以删掉视图函数中对文章列表中返回结果进行排序的代码了。

2、完善跳转链接

导航栏有一个 Black & White 的 Logo，我们希望点击它就能回到首页面，只需修改一下超链接即可。打开 base.html，修改 Logo 处的超链接：

```
【templates/base.html】
<header id="site-header">
  <div class="row">
    <div class="logo">
      - <h1><a href="index.html"><b>Black</b> &amp; White</a></h1>
      + <h1><a href="{% url 'blog:index' %}"><b>Black</b> &amp; White</a></h1>
    </div>
    ...
  </div>
</header>
```

另外导航栏还有一个首页导航按钮，也希望点击它就能回到首页面，修改的任务作为练习。

当然还有一些跳转可以完善，比如：

点击分类跳转到分类页面

```
<span class="post-category">
  <a href="{% url 'blog:category'
post.category.pk %}">{{ post.category.name }}</a>
</span>
```

点击评论量就跳转到文章详情页的评论处

```
<a href="{% post.get_absolute_url %}#comment-
area">{{ post.comment_set.count }} 评论</a></span>
```

3、显示正确的评论量

有两处地方显示的评论量，显示评论量的方法很简单。回顾一下我们是如何获取某篇 `post` 的下的评论列表的？我们使用的是 `post.comment_set.all()`。`all` 方法返回该 `post` 关联的评论列表。此外模型管理器（`comment_set` 是一个特殊的模型管理器）还有一个 `count` 方法，返回的是数量，即 `post` 下有多少条评论，我们可以直接在模板中调用这个方法：`{{ post.comment_set.count }}`。将评论量替换成该模板变量就可以正确显示文章的评论数了。