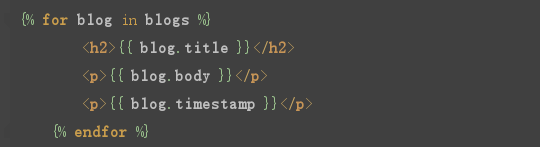
Django模板

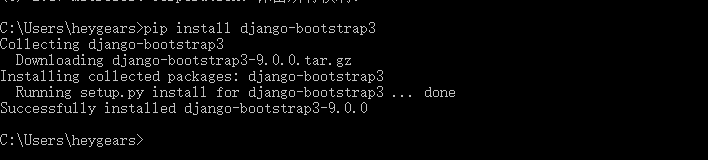
Django提供了模板语言，该模板语言主要控制着数据如何在页面中显示。例如，在本书第一章，将blog循环的显示在web页面上。



当然，如何使页面更漂亮，还是需要依靠前端技术HTML，css，JavaScript等

本章，带大家熟悉Bootstrap前端框架，为我们后面开发页面带来更好的操作。如果对Bootstrap不熟悉的话，可以看看关于Bootstrap的知识，入门很快。

现在我们通过命令行下载安装Bootstrap吧，打开cmd：



看到这里下载已经完成了。

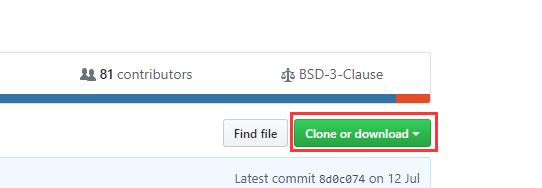
在Django的setting.py文件中的INSTALLED\_APPS模板中添加“bootstrap3”



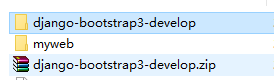
下载并运行demo

在django-bootstrap3项目中带了完整的demo，下面从github下载完整项目到本地

下载地址: github.com/dyve/django-bootstrap3

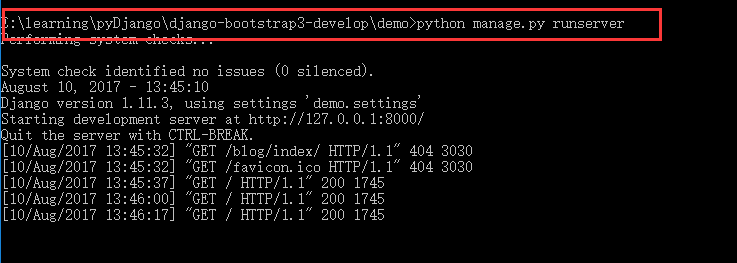


这方法是提供给电脑没有安装git的用户的，如果电脑安装了git的话，可以通过clone把项目克隆到本地。

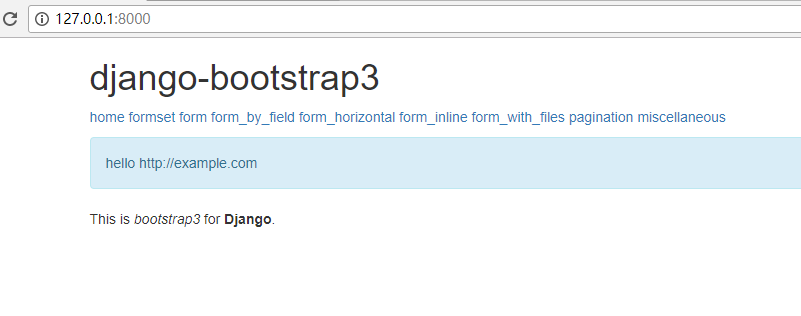


下载zip文件之后，解压到Python项目下，这样比较好找。进入demo目录下我们开始运行这个项目：

可能会有很多人一开始看到这条命令会以为是我们在运行Python项目的命令，其实我们要看清楚，我们是在bootstrap目录下运行的这个命令



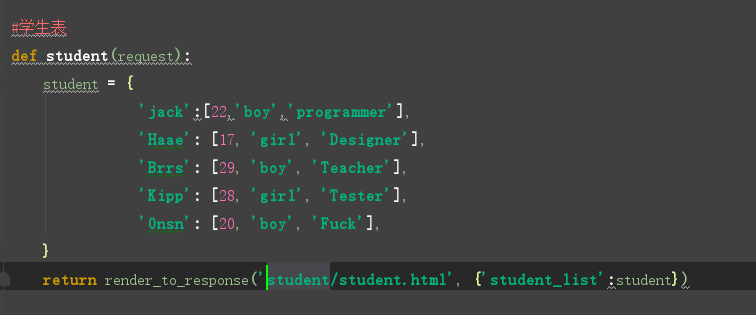
我们在浏览器中打开：<http://127.0.0.1:8000>（前面的http不能省，如果省去就会看不到css样式）



上面的菜单提供给我们不同样式的按钮、输入框、文本框、单选框、复选框、下拉框，甚至是翻页样式，我们点击上面的链接查看。就可以看到很多我们想看到的样式

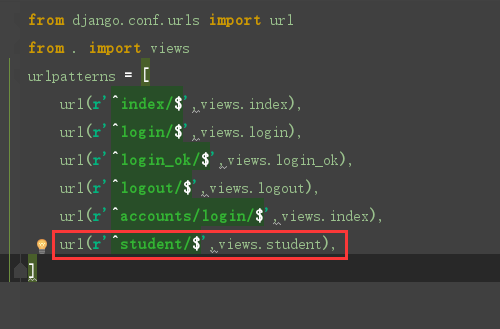
制作表格

首先我们回到myweb项目中，在/blog/目录下views.py文件中添加：



为了跟之前的blog文件区分开来，我们待会会新建一个student的文件夹

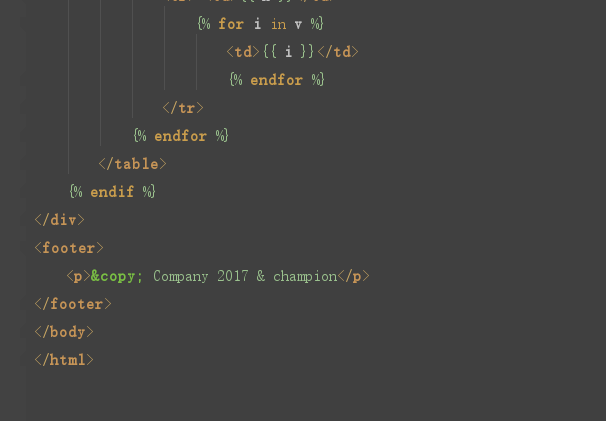
将我们新的视图链接配置到blog/urls.py中去



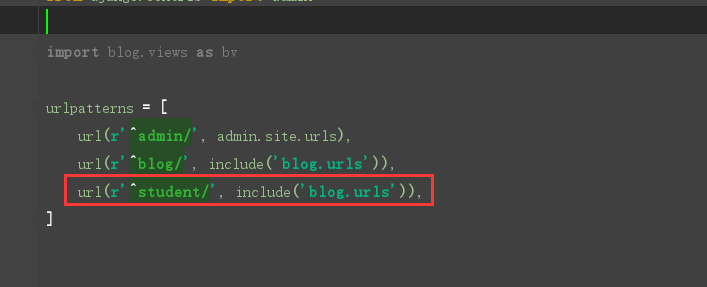
现在我们在templates中新建一个student文件夹，在文件夹里新建一个student.html文件

在html文件中，写入以下代码：

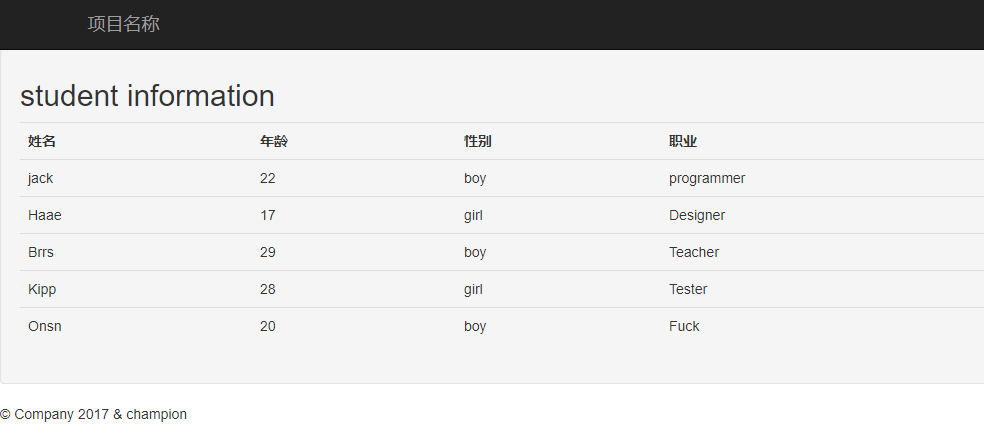




将我们的链接配置到django的url中去，打开myweb/myweb/urls.py文件：



现在我们可以通过django服务器访问了，启动django的内设server，然后访问: <http://127.0.0.1:8000/student/student/>

1.4 调用图书列表

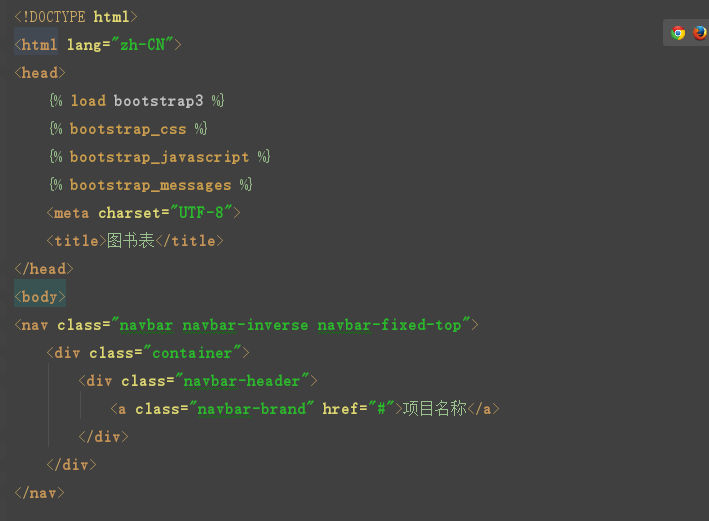
首先打开views.py文件，写入以下代码：

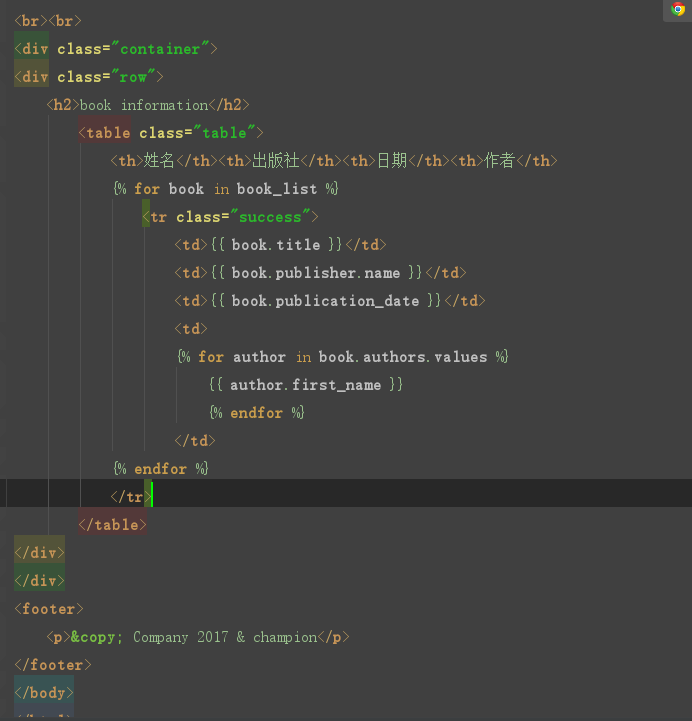


打开blog/urls.py



在blog/blog/下新建一个book.html文件，





保存，我们就可以去访问了：127.0.0.1/blog/book

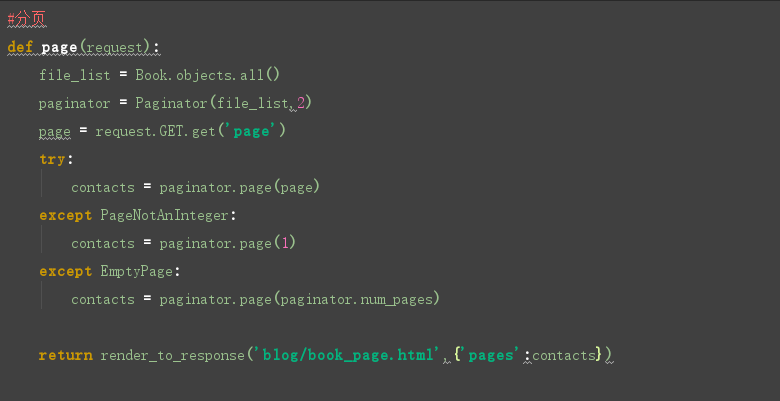


当我们的列表不断增加的时候，我们就需要使用下一个功能，就是分页功能。在很多页面，有不同的分页代码，包括php，js，Java...既然我们开发了列表，就应该离不开分页了。

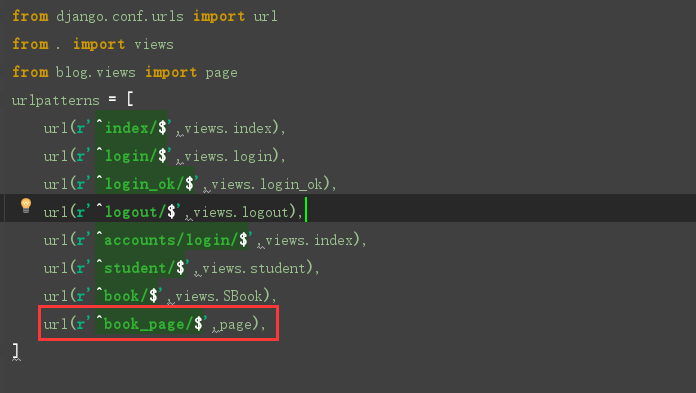
1.5 Pageinator类基本使用

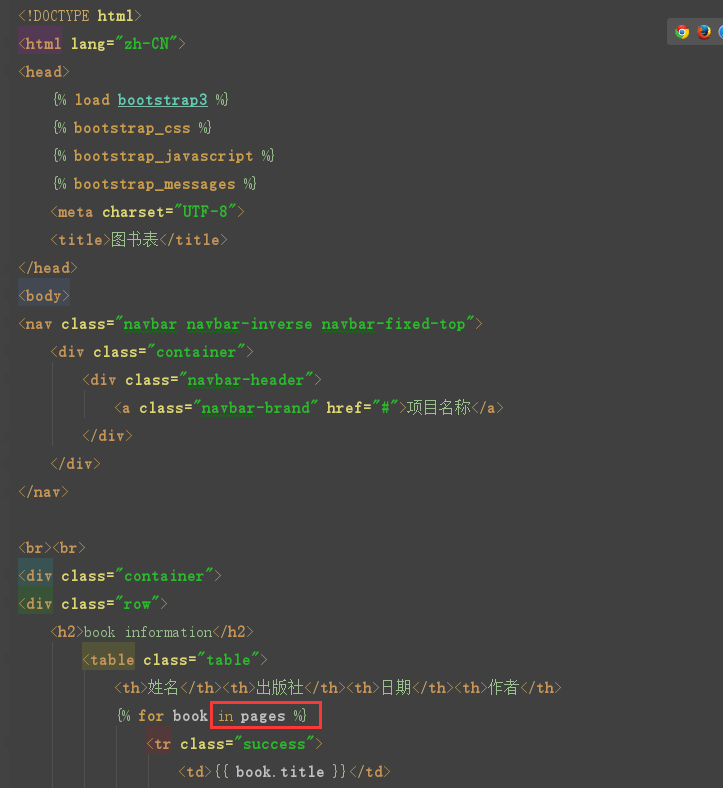
在django中，它就包含了Paginator分页类。所以，只需要调用他即可完成列表的分页功能了。

打开blog/views.py



将blog/urls.py文件添加url地址指向：



在blog/templates/blog目录下创建book\_page.html页面  






book\_page列表左下角显示分页功能，我们可以点击“next”和“previous”进行翻页

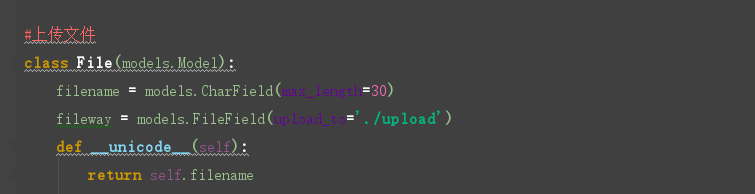


1.6 **文件上传功能**

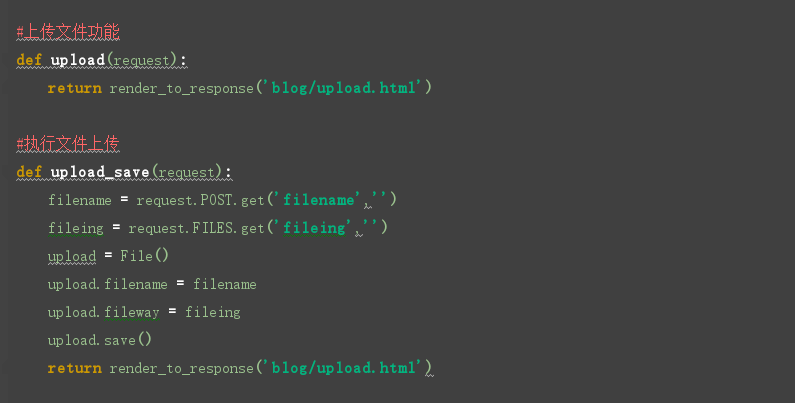
**开发上传功能**

创建模型：

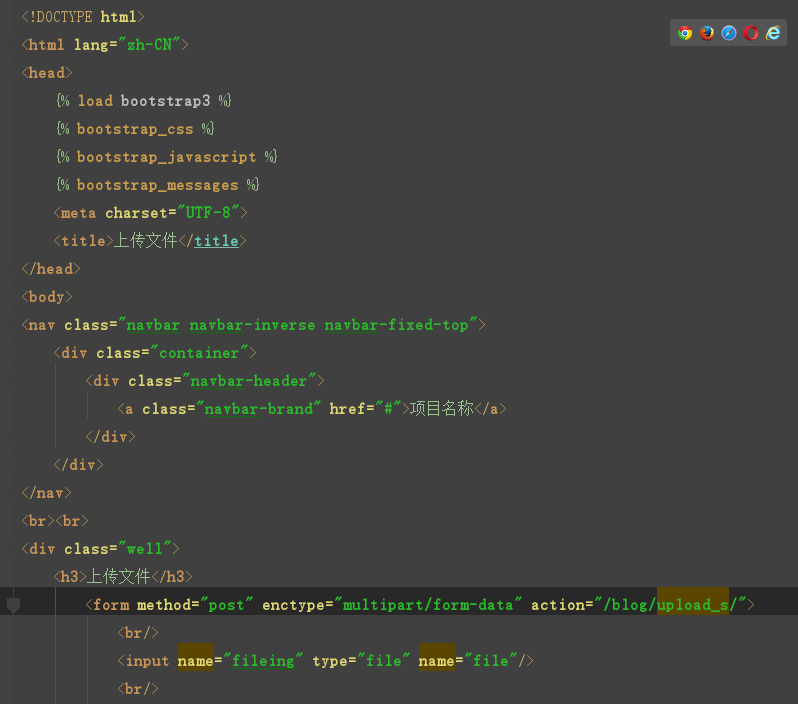
首先，我们要创建用于存放上传文件的表，打开blog/models.py文件，添加：

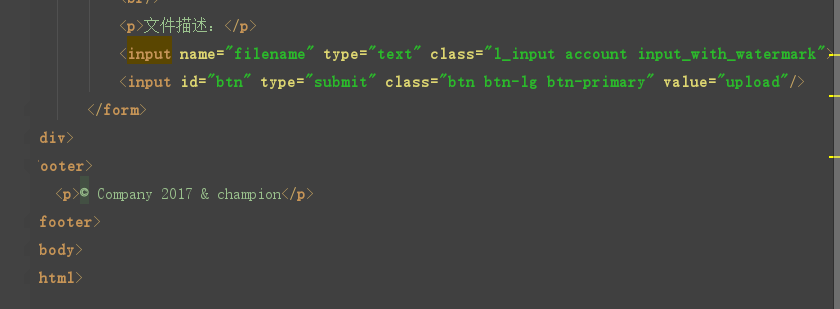


创建视图 ，在blog/views.py文件中增加：



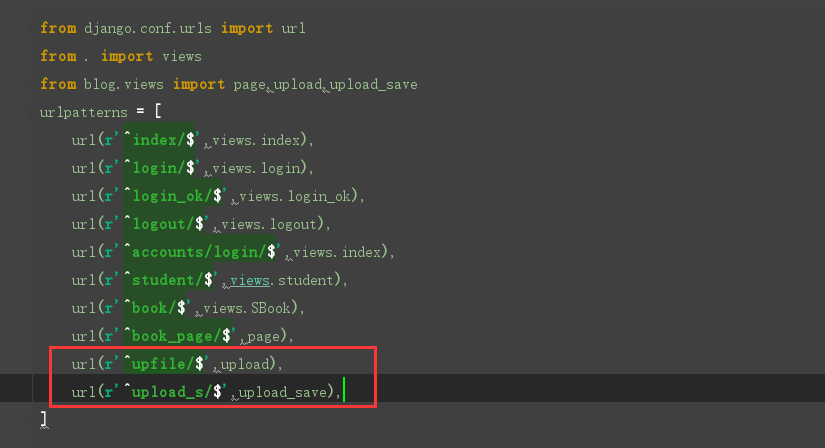
创建blog/templates/blog/upload.html页面



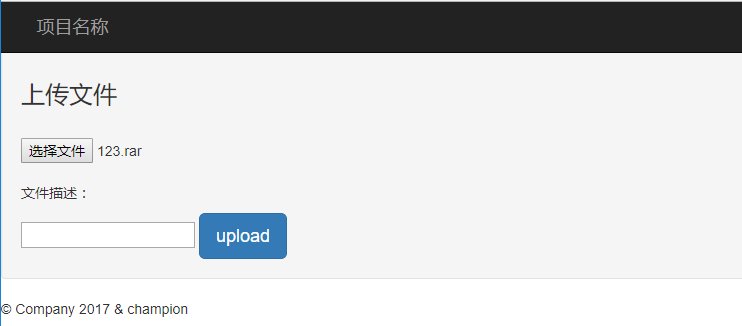


添加路径指向：

在blog/urls.py文件增加：



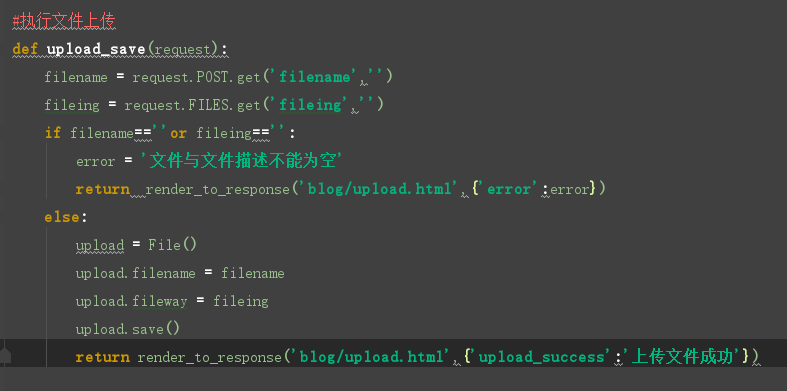
启动server，访问：127.0.0.1:8000/blog/upfile



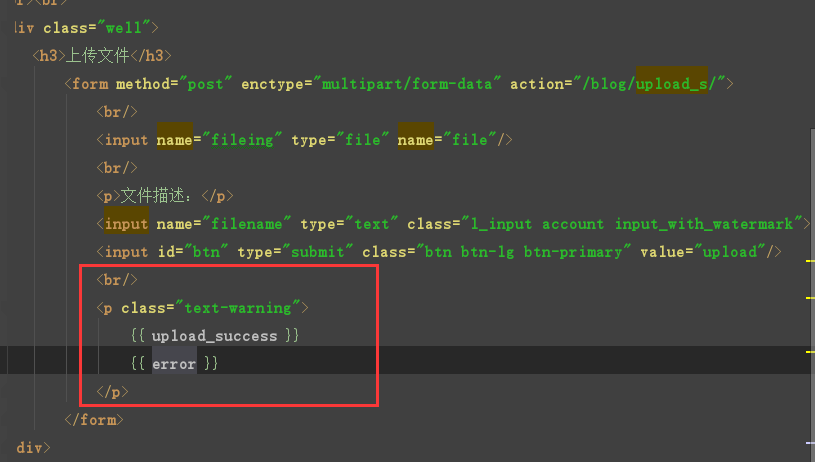
现在我们功能越来越多了。

但是现在还实现不了上传文件功能，只能点击文件。

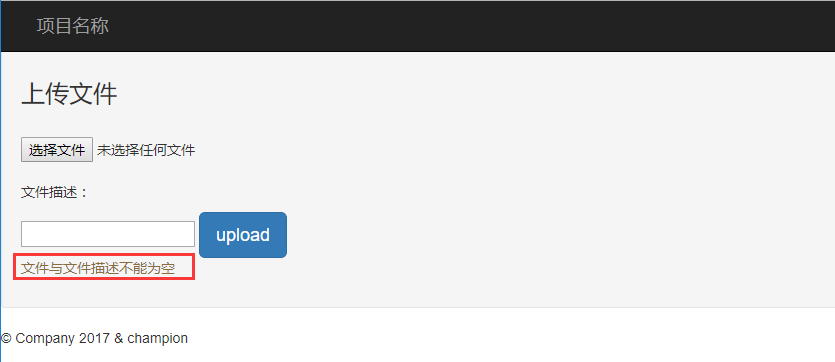
现在我们打开blog/views.py文件，修改upload\_s()逻辑

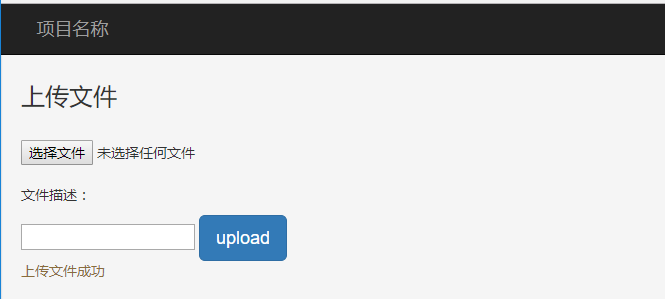


打开upload.html文件添加警告信息

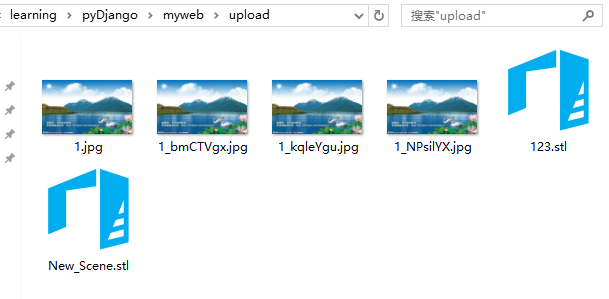


打开网页查看，





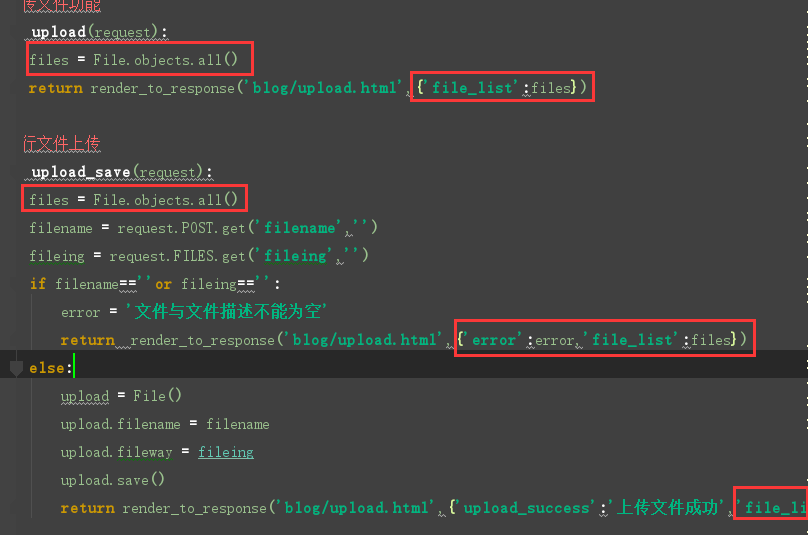
打开刚才创建的文件夹目录查看：

你会发现，刚才上传的文件都保存到upload目录下了。

现在我们制作上传文件列表

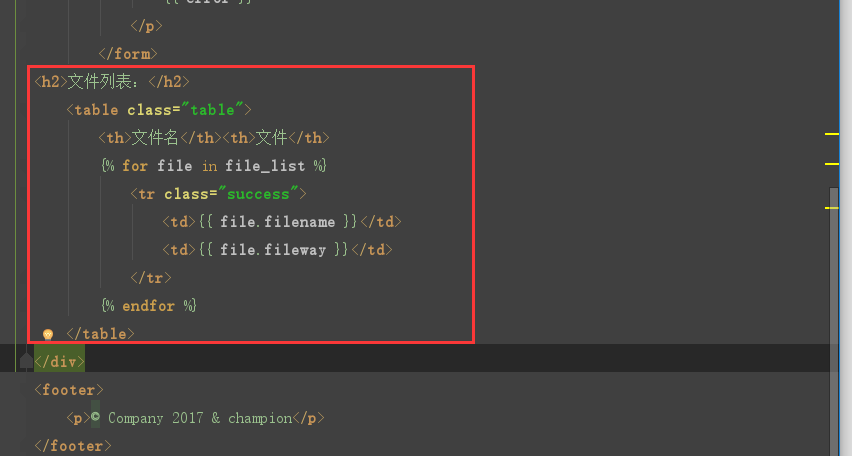
当我们想把上传文件的列表显示在页面上，我们就要：

打开blog/views.py文件：



在每个render\_to\_resounse里面添加同样的代码

在blog/templates/upload.html文件中添加：



刷新一下，显示得效果出来了：

