这一篇从 [教程第 5 部分](https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial05/) 结尾的地方继续讲起。在上一节中我们为网络投票应用编写了测试，而现在我们要为它加上样式和图片。

除了服务端生成的 HTML 以外，网络应用通常需要一些额外的文件——比如图片，脚本和样式表——来帮助渲染网络页面。在 Django 中，我们把这些文件统称为“静态文件”。

对于小项目来说，这个问题没什么大不了的，因为你可以把这些静态文件随便放在哪，只要服务程序能够找到它们就行。然而在大项目——特别是由好几个应用组成的大项目——中，处理不同应用所需要的静态文件的工作就显得有点麻烦了。

这就是 **django.contrib.staticfiles** 存在的意义：它将各个应用的静态文件（和一些你指明的目录里的文件）统一收集起来，这样一来，在生产环境中，这些文件就会集中在一个便于分发的地方。

自定义 *应用* 的界面和风格**[¶](https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial06/" \l "customize-your-app-s-look-and-feel" \o "永久链接至标题)**

首先，在你的 **polls** 目录下创建一个名为 **static** 的目录。Django 将在该目录下查找静态文件，这种方式和 Diango 在 **polls/templates/** 目录下查找 template 的方式类似。

Django 的 [**STATICFILES\_FINDERS**](https://docs.djangoproject.com/zh-hans/2.1/ref/settings/#std:setting-STATICFILES_FINDERS) 设置包含了一系列的查找器，它们知道去哪里找到 static 文件。**AppDirectoriesFinder** 是默认查找器中的一个，它会在每个 [**INSTALLED\_APPS**](https://docs.djangoproject.com/zh-hans/2.1/ref/settings/#std:setting-INSTALLED_APPS) 中指定的应用的子文件中寻找名称为 **static** 的特定文件夹，就像我们在 **polls** 中刚创建的那个一样。管理后台采用相同的目录结构管理它的静态文件。

在你刚创建的 **static** 文件夹中创建一个名为 **polls** 的文件夹，再在 **polls** 文件夹中创建一个名为 **style.css** 的文件。换句话说，你的样式表路径应是 **polls/static/polls/style.css**。因为 **AppDirectoriesFinder** 的存在，你可以在 Django 中简单地使用以 **polls/style.css** 的形式引用此文件，类似你引用模板路径的方式。

**静态文件命名空间**

虽然我们 *可以* 像管理模板文件一样，把 static 文件直接放入 **polls/static** ——而不是创建另一个名为 **polls** 的子文件夹，不过这实际上是一个很蠢的做法。Django 只会使用第一个找到的静态文件。如果你在 *其它* 应用中有一个相同名字的静态文件，Django 将无法区分它们。我们需要指引 Django 选择正确的静态文件，而最简单的方式就是把它们放入各自的 *命名空间* 。也就是把这些静态文件放入 *另一个* 与应用名相同的目录中。

将以下代码放入样式表(**polls/static/polls/style.css**)：

polls/static/polls/style.css[**¶**](https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial06/#id1)

**li** **a** {

**color**: **green**;

}

下一步，在 **polls/templates/polls/index.html** 的文件头添加以下内容：

polls/templates/polls/index.html[**¶**](https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial06/#id2)

{% **load** static %}

<**link** rel="stylesheet" type="text/css" href="{% **static** 'polls/style.css' %}">

**{% static %}** 模板标签会生成静态文件的绝对路径。

以上就是你全部需要做的事。浏览器重载 **http://localhost:8000/polls/**，然后你可以看到投票链接是绿色的（Django 样式），这意味着你的样式表被正确加载了。

添加一个背景图**[¶](https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial06/" \l "adding-a-background-image" \o "永久链接至标题)**

接着，我们会创建一个用于存在图像的目录。在 **polls/static/polls** 目录下创建一个名为 **images**的子目录。在这个目录中，放一张名为 **background.gif** 的图片。换言之，在目录 **polls/static/polls/images/background.gif** 中放一张图片。

随后，在你的样式表（**polls/static/polls/style.css**）中添加：

polls/static/polls/style.css[**¶**](https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial06/#id3)

**body** {

**background**: **white** url("images/background.gif") **no-repeat**;

}

浏览器重载 **http://localhost:8000/polls/**，你将在屏幕的左上角见到这张背景图。

**警告**

当然，**{% static %}** 模板标签在静态文件（例如样式表）中是不可用的，因为它们不是由 Django 生成的。你仍需要使用 *相对路径* 的方式在你的静态文件之间互相引用。这样之后，你就可以任意改变 **STATIC\_URL`（由 :ttag:`static** 模板标签用于生成 URL），而无需修改大量的静态文件。

这些只是 **基础** 。更多关于设置和框架的资料，参考 [静态文件解惑](https://docs.djangoproject.com/zh-hans/2.1/howto/static-files/) 和 [静态文件指南](https://docs.djangoproject.com/zh-hans/2.1/ref/contrib/staticfiles/)。[部署静态文件](https://docs.djangoproject.com/zh-hans/2.1/howto/static-files/deployment/) 介绍了如何在真实服务器上使用静态文件。

当你熟悉静态文件后，阅读 [此教程的第 7 部分](https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial07/) 来学习如何自定义 Django 自动生成后台网页的过程。