

一、静态网页及资源拷入对应目录，并对页面js,css,images做引用路径的修改

- 静态文件拷贝到templates文件下
- 静态资源拷贝到static文件下
- 修改js,css,image资源引用路径

File explorer showing project structure: templates, static, media, etc.

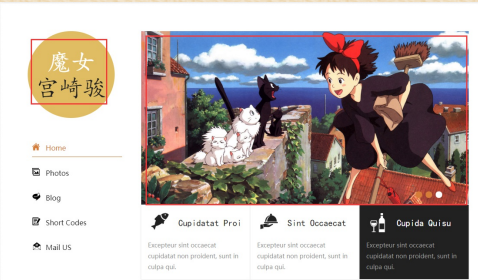
```
<!DOCTYPE html>
<html>
<head>
<title>Home</title>
<!-- For mobile view -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="keyword" content="Great Taste Responsive web template, Bootstrap Web Template, Responsive web template, Free webdesigns for Nokia, Samsung, LG, SonyEricsson, Acer, Apple IOS, Google Chrome, etc." />
<!-- For mobile view -->
<script>$(function(){
function hideNavbar() {
if (window.scrollY > 0) {
}
}
});
</script>
<link href="static/css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />
<link href="static/css/style.css" rel="stylesheet" type="text/css" media="all" />
<!-- For mobile view -->
<script src="static/js/jquery-2.1.1.min.js"></script>
</head>
<body>
<!-- Banner body -->
<div class="banner-body">
```

二、替换页面标题及轮播图片

File explorer showing project structure: templates, static, media, etc.

```
<!-- Banner body -->
<div class="banner-body">
<div class="container">
<div class="banner-body-content">
<div class="col-xs-8 banner-body-left">
<div class="logo">

</div>
<div class="top-nav">
<div class="navbar navbar-default">
<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
</button>
</div>
<div class="collapse navbar-collapse">
<ul class="list-unstyled">
<li><a href="static/images/logo.png" alt="Logo" /></a></li>
</ul>
</div>
</div>
</div>
</div>
</div>
```



三、菜单动态数据库读取

1、创建表

models文件，定义表结构

```
from django.db import models

# Create your models here.

class Nav (models.Model):

    name=models.CharField(max_length=10,unique=True,verbose_name="导航名称") #int可以不加长度，字符串类型必须加长度

    url=models.CharField(max_length=100,unique=True,verbose_name='导航链接')

    classname=models.CharField(max_length=50,unique=True,verbose_name='导航图标')

    is_delete=models.SmallIntegerField(default=1,verbose_name='是否被删除')

    create_time=models.DateTimeField(verbose_name='创建时间',auto_now_add=True) #auto_now_add插入数据自动取当前时间

    update_time=models.DateTimeField(verbose_name='修改时间',auto_now=True)#auto_now 修改时间自动取当前时间


    def __str__(self):

        return self.name


class Meta:

    verbose_name='导航表'

    verbose_name_plural=verbose_name

    db_table='nav' #指定表名

    ordering=['update_time'] #查询数据的时候，用来排序的
```

2、生成表

命令行输入

1、python manage.py makemigrations ---创建表结构

2、python manage.py migrate ---创建数据库表

3、创建成功，可通过Navicat的SQLite方式连接数据库

创建连接选择项目中的db.sqlite3文件，进行建连接，连接后，查看新建表

表

auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
django_admin_log
django_content_type
django_migrations
django_session
nav
sqlite_sequence

4、为表加数据

models_test.py文件加数据

```
import django,os

os.environ.setdefault('DJANGO_SETTINGS_MODULE','django_study2.settings') #设置django的数据库配置文件

django.setup()

from user import models

#新增

models.Nav.objects.create(name='首页',url='index',classname='home') #新增第一种方法

models.Nav.objects.create(name='图集',url='photos',classname='picture1')

models.Nav.objects.create(name='相关新闻',url='blog',classname='edit1')

models.Nav.objects.create(name='剧情介绍',url='short-codes',classname='text-size1')

models.Nav.objects.create(name='读者来信',url='mail',classname='envelope1')
```

id	name	url	classname	is_delete	create_time	update_time
1	首页	index	home	1	2019-10-16 14:3	2019-10-16 14:36
2	图集	photos	picture1	1	2019-10-16 14:3	2019-10-16 14:36
3	相关新闻	blog	edit1	1	2019-10-16 14:3	2019-10-16 14:36
4	剧情介绍	short-codes	text-size1	1	2019-10-16 14:3	2019-10-16 14:36
5	读者来信	mail	envelope1	1	2019-10-16 14:3	2019-10-16 14:36

5、view文件中获取数据库数据，并返回

```
from django.shortcuts import render

from user import models

from django.shortcuts import HttpResponseRedirect

# Create your views here.

def index(request):

    tab =models.Nav.objects.filter(is_delete=1) #获取数据库数据

    dic={"tab":tab}

    return render(request, 'index.html',dic) #字段方式传参
```

6、把index加入到urls文件中

```
from django.contrib import admin

from django.urls import path

from user import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('index/', views.index),

]
```

7、对index.html文件中的内容做动态替换


```
<div class="collapse navbar-collapse nav-wil" id="bs-example-navbar-collapse-1">

    <nav class="stroke">

        <ul class="nav navbar-nav">

            {% for k in tab %}

                {% if k.url == 'index' %}

                    <li class="active"><a href="{(k.url)}.html">i class={ (k.classname)}>

</i>{(k.name)}</a></li>

                {% else %}

                    <li><a href="{(k.url)}.html" class="hvr-underline-from-left">i class={ (k.classname)}>

</i>{(k.name)}</a></li>

                {% endif %}

            {% endfor %}

        </ul>

    </nav>

</div>
```

8、启动manage.py文件，浏览器访问http://127.0.0.1:8000/index



四、模块化

1、抽取公共部分

比如左边栏，底部及右上半栏每个页面都是一样的，仅仅右下半栏每个页面不一致，新建base.html文件，中间变动的部分用如下代码代替

```
{% block content %}

{% endblock %}
```




```
from user import models

def nav_title_process(request):

    '''上下文管理器，这个函数里面返回的每个变量，在每个页面里面都可以用'''

    title = models.top3.objects.filter(is_delete=1)

    content={"title":title}

    return content
```



3、在html文件中引用该变量



```
<div class="banner-bottom">

    {% for info in title %} #循环变量中的title

    <div class="col-md-4 banner-left">

        <div class="col-xs-3 banner-left1">

            <div class="banner-left1">

                <span> </span>

            </div>

        </div>

        <div class="col-xs-9 banner-right1">

            <h3>{{ info.title }}</h3> #读取info中的文章标题

        </div>

        <div class="clearfix"> </div>

        <p>{{ info.content }}</p> #读取info中的文章内容

        </div>

    {% endfor %} #结束循环

    <div class="clearfix"></div>

</div>

</div>
```



4、页面效果，如下

