

What are Django static files?

- Static files are files that don't change when your application is running.
- These files do a lot to improve your application, but they aren't dynamically generated by your Python web server.
- In a typical web application, your most common static files will be the following types: Cascading Style Sheets(CSS), JavaScript and Images.
- Web applications generally need to serve additional files — such as images, JavaScript, or CSS — necessary to render the complete web page.
- In Django, we refer to these files as “static files”.

How to Managing Static files

We have to create some settings in **settings.py** file

- Django provides the default application about staticfiles, the application name is **'django.contrib.staticfiles'**, which is available under **INSTALLED_APPS** automatically.
- Create **STATIC_DIR** variable like **STATIC_DIR = os.path.join(BASE_DIR, 'static')**
- Note: This **STATIC_DIR** variable should be created after **BASE_DIR** line in settings.py file
- Create **STATICFILES_DIRS** variable at the end of settings.py file, like below
STATICFILES_DIRS = [STATIC_DIR]
- Create a folder like static inside our root folder location and maintain required files here for displaying in our application.

Create a Project to display Static Files using Django :

Step1: Create a Django ProjectName like **StaticFiles_Project**

Step2: Create a Application Name like **StaticFiles_App**

Step3: Create Database Name like **7am_staticfilesdb**

Step4: Open settings.py do like below

----->> Add our applicationName inside **INSTALLED_APPS** section

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'StaticFiles_App.apps.StudentappConfig',  
]
```

----->> Goto settings.py file and configure database details under **DATEBASE** section.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': '7am_inheritedb',
        'USER': 'root',
        'PASSWORD': 'root',
    }
}
```

----> Goto to **TEMPLATE** section and configure the templates path directory like below.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'] ,
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

Note1: `django.contrib.staticfiles` default application should be available under **INSTALLED_APPS**

```
INSTALLED_APPS = [
    '-----',
    '-----',
    'django.contrib.staticfiles',
    '-----',
]
```

Note2: Create `STATIC_DIR` variable under `BASE_DIR`, like
STATIC_DIR = os.path.join(BASE_DIR , 'static')

Note 3: Create `STATICFILES_DIRS` variable at the end of settings.py file
STATICFILES_DIRS = [STATIC_DIR]

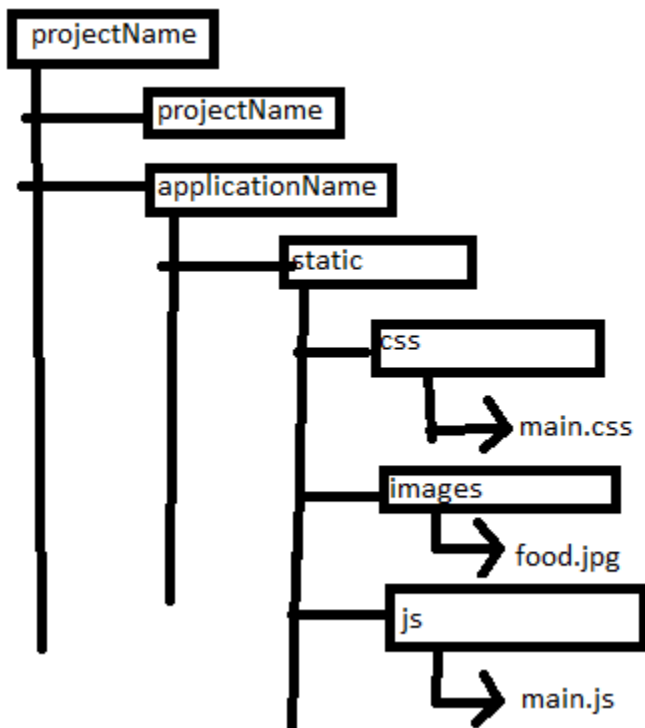
Step6: Create a folder with name '**static**' inside our applicationName.

Step7: Create a folder inside static folder and name as '**css**' for css files storing purpose.

Step8: Create a folder inside static folder and name as '**images**' for images storing purpose.

Step9: Create a folder in static folder and name as 'js' for javascript files storing purpose.

For example



Step10: Create required files in required folders

- Create one css file in css folder and name as home.css
- Copy and paste any one image in images folder and names as food.jpg
- Create one javascript file in js folder and name as main.js

Step11: Goto views.py file and write the following view

```
from django.shortcuts import render
```

```
def homeView(request):
    return render(request, 'statics.html')
```

Step12: Create statics.html file in templates folder

- **Note :** To display any static files within an app, at the top of the template page, you must be load static folder using load tag like bellow line.

```
{% load static %}
```

- Now use required files location using static tag structure with required file tags.

For example: `
```

```
<head>
```

```

 <title>Static Files</title>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css">
 <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
 <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js"><
/script>
 <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"></scrip
t>

<script src="{% static 'js/main.js' %}" type="text/javascript"></script>

<link rel="stylesheet" href="{% static 'css/main.css' %}">

</head>
<body>
 <h1>Welcome to Django Restaurant</h1>

</body>
</html>

```

**Step13: open urls.py and create required url to execute views.**

```

urlpatterns = [
 path("home/", views.homeView, name="home")
]

```

**Step14: Now run the server using runserver command.**

**Step15 : Send the requested url from browser and get response now.**