mysite/

manage.py

mysite/

\_\_init\_\_.py

settings.py

urls.py

asgi.py

wsgi.py

these are files:

* The outer **mysite/** root directory is a container for your project. Its name doesn’t matter to Django ; You can rename it to anything you like
* **Manage.py :**A command-line utility that lets you interact with this Django project in Various ways . You can read all the details about **manage.py in djnago-admin and manage.py** 
  + - **The command-line utility is an example of an application that uses the service interface to run and manage services**
    - **A utility called a compiler is program which translates the HLL program into a form understandable to the hardware .**
* The inner **mysite/** directory is the actual file that tells python package for your project . its name is the python package name you’ll need to use to import anything inside it (eg:: **mysite.urls)**
* **Mysite/\_\_init\_\_.py :**An empty file that tells Python that this directory should be considered a Python package . if you’re a python beginner , read more about packages in the official Python docs
* **Mysite/setting.**py :settings/configuration for this Django project . **Django settings** will tell about how setting work
* **Mysite/urls.py :**The URL declarations for this Django project; a “table of contents” of your Django -powered site .
  + - A clean , elegant URL scheme is an important detail in a high-quality web application. Django lets you design URLs however you want , with no framework limitations .
    - **HOW DJANGO PROCESSES A REQUEST**
      * + When a user request a page from your Django-powered site, this is the algorithm the system follows to determine which python code to execute.
        + 1) Django determines the root URL conf module to use. Ordinaly, this is the value o the ROOT\_URLCONF setting , but if the incoming **HTTPREQUEST** object has a **urlconf** attribute (set by middleware), its value will be used in place of the **ROOT\_URLCONF** setting .
        + 2) Django loads that python module and looks for the varuable **urlpatterns.** This should be a sequence of **Django.urls.path()** and /or **Django.urls.re\_path()** instances

Django.urls functions for use in URLconfs

Path()

Path(route, view,kwargs=None, name None)

* **Mysite/asgi.py::** An entry-point for ASGI-compatible web servers to server to serve your project
  + HOW TO DEPLOY WITH ASGI
    - As well as WSGI , Django also supports deploying on ASGI, the emerging Python standard for asynchronous web servers for your project, and direct any ASGI-COMPLIANT application server to use .
    - THE APPLICATIOON OBJECT
      * Like WSGI,ASGI has you supply an application callable which the application server uses to communicate with your code. Its commonly provided as an object name **application**  in python module accessible to the server
* **Mysite/wsgi.py**  an entry-point for WSGI-compatible web servers to serve your project

**DATABASE SETUP**

Now open up **mysite/settings.py** . its normal Python module with module-level variable representing Django settings.

By default, the configuration uses SQLite . if you’re new to databases, or you’re just interested in trying Django , this is the easiest choice . SQLite is included in Python, so won’t need to install anything else to support your database. When starting your first real project, however , you may want to use a more scalable database like PostgreSql to avoid database-switching headaches down the road .

If you wish to use another database, install the appropriate database bindings and change the following keys in the DATABASES ‘default’ item to match your database connection settings