---------- THE ANATOMY OF A DJANGO APP ----------

models.py

A Django model is an object that maps your database data without SQL. We are defining that we want to store a title, which will be defined as a CharField, which is a small string of data. We are also saying that we don't want that string to be longer then 200 characters.

We also define that we want to store our article text in a TextField which is used to store large strings of data.

There are many types of data you can store including strings, dates, times, numbers of many different sizes, as well as validated file, image, URL, IP Address and emails data types.

home.html

home.html will be in your template file folder. Templates are HTML files. Django separates code and design. This is so designers can work on the HTML, CSS, images and user interaction without any knowledge of Python or SQL.

Your templates folder will be on the same level as your project and apps folders. You'll tell your settings file where your templates folder is by setting the DIRS part of the TEMPLATES list with os.path.join(BASE_DIR, 'templates') which just means it is in the base directory and it is named templates.

Your images will be in the base directory as well and this folder is normally named media. Some people put CSS and images in the same static folder. You tell the settings folder where that is with this command MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

Your CSS files will be in a folder normally called static/css and you tell settings about it with STATICFILES_DIRS = (os.path.join('static'),)

You'll commonly also use a base.html file that you'll then inherit for all of your other pages. In the base.html page you then define the parts that will change for the files that inherit from it with the tags {% block <name> %} and {% endblock <name> %}

You can pull data from your model with tags like {{ article.text }} where the object name is article and the object field is named text.

views.py

The view takes data from the defined model (database) and sends it to the template (HTML). Django has many types of views. A ListView is used when you want to provide a list of objects. In this situation that would be a list of articles.

context_object_name is assigned the name you wish to use in your template to refer to the article list you wish you display.

project/urls.py

This file defines site wide navigation. You don't want to have all your URLs in one file because if 2 apps have the same name for a view that would cause a conflict.

To solve this issue we define a urls file for each app we create. We have a articles/urls.py file where we say if the user goes to http://127.0.0.1:8000/ project/urls.py says the articles app will handle that request. It includes here the articles/urls.py information that then tells the browser to load the HomePageView class and display it.

settings.py

Defines the location of the app configuration class in the INSTALLED_APPS list. So if the file is in articles/apps.py/ArticlesConfig class you'd enter articles.apps.ArticlesConfig in the list

apps.py

Used for application configuration. Here we define the name of the app for the admin interface.

articles/urls.py

This file is called when the project urls file says that url requests can be resolved here. This file states that if the user requests the base directory that we should display the template listed in the provided view along with the model data defined there.