**Django Python**

Django assumes that all websites are a collection of Apps.

To start a project>

Django-admin startproject “yoursitename”

This creates a “mysite” folder with a mysite folder in it. It has a bunch of python files in it.

There is also a file called manage.py.

Open up a cmd at that file location and call;

>>py manage.py startapp main (This starts another app called main. Remember that Django identifies collections of Apps. ESSENTIALLY AN APP IS A CONTROLLER)

To start the server;

>>py manage.py runserver

**How routing works?**

When someone comes to a particular controller (“app” in Django terms), this results in the system looking at urls.py to check if that url pattern exists(method exists for that controller)

**Models**

Every model is going to inherit from Django’s model class.

Each Model we create will be considered as a Table in the SQLITE3 Database. Each attribute of a model will be a column in a table. We need to 1) Make Migrations and 2) Migrate

1. Make Migrations

>> py manage.py makemigrations

The above will give “No Changes Detected”. This is because although we can point to the new apps(controllers) we have, we haven’t actually installed them.

**Installing Apps**

In settings.py, under Installed Apps, add ‘app\_name.apps.MainConfig’

You can see what was done in the migrations folder.

To see the sqlite command that was carried out:

>> py manage.py sqlmigrate main 0001

CREATE TABLE "main\_tutorial" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "tutorial\_title" varchar(200) NOT NULL, "tutorial\_content" text NOT NULL, "tutorial\_published" datetime NOT NULL);

COMMIT;

1. Migrate

>>py manage.py migrate

**Creating an Admin**

>> python manage.py createsuperuser

* Then we access this at “server\_address/admin”

To see the models we create, we need to register our models

To Register our models we do it in **admin.py**:

>> admin.site.register(Tutorial)

Then we can see this in “server\_address/admin”. But if we want to change anything, like the order in which it is presented, we create a class in admin.py as:

Class TutorialAdmin(admin.ModelAdmin):

Fields = [“tutorial\_title”,”tutorial\_content”,”tutorial\_published”]

Then we register both Tutorial(our model class) and TutorialAdmin: admin.site.register(Tutorial,TutorialAdmin)

**WHEN YOU MAKE CHANGES TO YOUR MODEL, YOU NEED NOT DO MIGRATIONS AGAIN**

**Putting in an editor to your textboxes:**

We can do this by using a django library called tinymce4. After doing pip install Django-tinymce4-lite, we need to install it into our installed apps in **settings.py.**