**🛠️ 一、環境準備**

1. **安裝 Python**
   * 先確認有安裝 Python 3.10 以上版本：
   * python --version

或

python3 --version

* + 若沒有，請到 [Python 官網](https://www.python.org/downloads/) 下載安裝。

1. **安裝 VS Code**
   * 下載 Visual Studio Code
   * 在 VS Code 安裝以下延伸套件：
     + **Python** (Microsoft 官方)
     + **Pylance**
     + **Django** (可選)
2. **建立虛擬環境 (virtual environment)**  
   在專案資料夾下執行：
3. python -m venv venv

啟動虛擬環境：

* + Windows:
  + venv\Scripts\activate
  + macOS/Linux:
  + source venv/bin/activate

**🚀 二、建立 Django 專案**

1. **安裝 Django**
2. pip install django
3. **建立專案**
4. django-admin startproject myproject
5. cd myproject
6. **啟動開發伺服器**
7. python manage.py runserver

在瀏覽器打開：  
👉 <http://127.0.0.1:8000>

如果看到 Django 預設的歡迎畫面，代表成功 🎉

**📂 三、建立 App (功能模組)**

在 Django 中，專案底下會有多個 App，各自負責一個功能。

1. 建立 app
2. python manage.py startapp myapp
3. 在 settings.py 的 INSTALLED\_APPS 加入：
4. INSTALLED\_APPS = [
5. ...
6. 'myapp',
7. ]
8. 在 myapp/views.py 定義一個簡單的頁面：
9. from django.http import HttpResponse
10. def home(request):
11. return HttpResponse("Hello Django!")
12. 在 myproject/urls.py 加入路由：
13. from django.contrib import admin
14. from django.urls import path
15. from myapp import views
16. urlpatterns = [
17. path('admin/', admin.site.urls),
18. path('', views.home), # 根目錄顯示 Hello Django!
19. ]
20. 再次啟動伺服器：
21. python manage.py runserver

打開 <http://127.0.0.1:8000/> 就會看到 **Hello Django!**

**🛠️ 四、VS Code 提升開發體驗**

1. **設定 Debug**  
   在 .vscode/launch.json 新增：
2. {
3. "version": "0.2.0",
4. "configurations": [
5. {
6. "name": "Django",
7. "type": "python",
8. "request": "launch",
9. "program": "${workspaceFolder}/manage.py",
10. "args": [
11. "runserver",
12. "8000"
13. ],
14. "django": true
15. }
16. ]
17. }

之後可以直接在 VS Code 按 F5 執行。

1. **格式化 & Linting**
   * 安裝 black 或 autopep8：
   * pip install black
   * VS Code 設定 "python.formatting.provider": "black"

**✅ 五、下一步可以嘗試的功能**

* 使用 **Django ORM** 連接資料庫 (SQLite 內建)
* 建立 Model + Migration
* 使用 Django Template 顯示 HTML 頁面
* 建立 REST API（可用 Django REST Framework）

要不要我幫你整理一份 **最小可執行的 Django 範例專案程式碼結構**（包含 urls.py、views.py、settings.py 基本修改），讓你複製就能跑？

**你說：**

使用 Django ORM 連接資料庫 (SQLite 內建) 請仔細說明步驟

**ChatGPT 說：**

已思考 1m 5s

沒問題！下面是一份「用 Django ORM 操作 SQLite（內建）」的完整、實作型教學。你只要照步驟做，就能在 VS Code 裡把資料表建起來、遷移（migrate）、在 Admin 新增資料，並用 ORM 查詢與更新。

**1) 確認資料庫設定（SQLite）**

Django 預設就是 SQLite，不用額外安裝資料庫伺服器。settings.py 會長這樣（重點是 DATABASES）：

# myproject/settings.py

from pathlib import Path

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

DATABASES = {

"default": {

"ENGINE": "django.db.backends.sqlite3",

"NAME": BASE\_DIR / "db.sqlite3",

}

}

第一次執行 migrate 後，專案根目錄會出現 db.sqlite3 檔案。

**2) 建立 App 與 Model**

如果你還沒有 App，先建一個（範例用 blog）：

python manage.py startapp blog

把 blog 加進 settings.py 的 INSTALLED\_APPS：

INSTALLED\_APPS = [

# ...

"blog",

]

建立一個示範的資料表（Post 貼文），展示欄位、索引、唯一性與關聯到 User：

# blog/models.py

from django.db import models

from django.contrib.auth import get\_user\_model

class Post(models.Model):

STATUS\_CHOICES = [

("draft", "草稿"),

("pub", "發布"),

]

title = models.CharField(max\_length=200, db\_index=True)

body = models.TextField()

author = models.ForeignKey(

get\_user\_model(), on\_delete=models.CASCADE, related\_name="posts"

)

status = models.CharField(

max\_length=5, choices=STATUS\_CHOICES, default="draft", db\_index=True

)

created\_at = models.DateTimeField(auto\_now\_add=True, db\_index=True)

updated\_at = models.DateTimeField(auto\_now=True)

class Meta:

ordering = ["-created\_at"]

indexes = [

models.Index(fields=["status", "created\_at"]),

]

constraints = [

models.UniqueConstraint(

fields=["title", "author"], name="uniq\_title\_author"

)

]

def \_\_str\_\_(self):

return self.title

**3) 產生並套用遷移（Migrations）**

# 讀取 models.py 變化，產生遷移檔

python manage.py makemigrations

# 套用到資料庫（建立/更新資料表）

python manage.py migrate

執行後 db.sqlite3 就完成了，且包含 blog\_post 等資料表。

**4) Django Admin 管理資料**

註冊 Model 到 Admin：

# blog/admin.py

from django.contrib import admin

from .models import Post

@admin.register(Post)

class PostAdmin(admin.ModelAdmin):

list\_display = ("id", "title", "author", "status", "created\_at")

list\_filter = ("status", "created\_at", "author")

search\_fields = ("title", "body")

建立管理者帳號與登入：

python manage.py createsuperuser

python manage.py runserver

開瀏覽器到 http://127.0.0.1:8000/admin/ 登入後，就能用後台新增/編輯 Post。