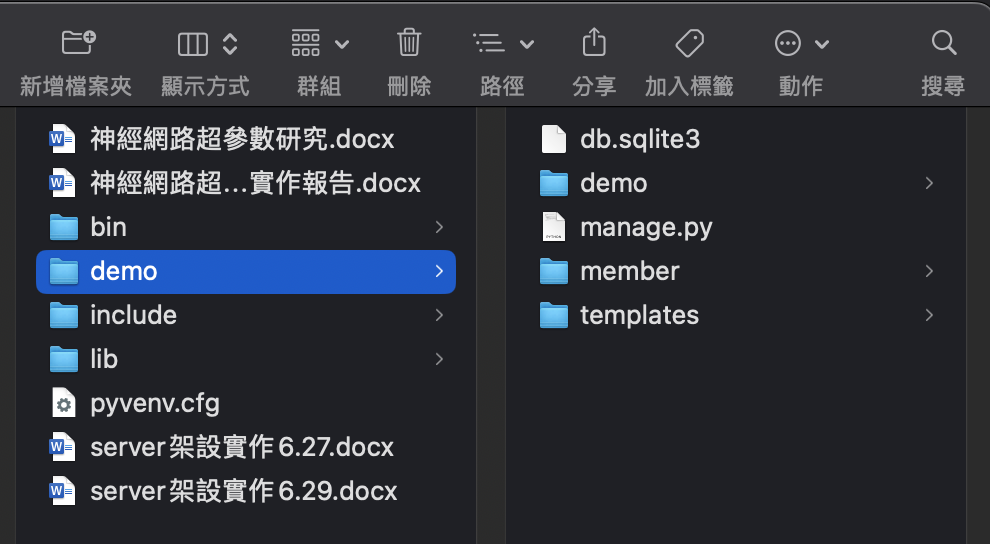
先前往要新增Django檔案的資料夾位址輸入：Django-admin startproject demo後會新增一個名叫demo的原資料夾，此資料夾內部還有一個同名資料夾做為主資料夾。

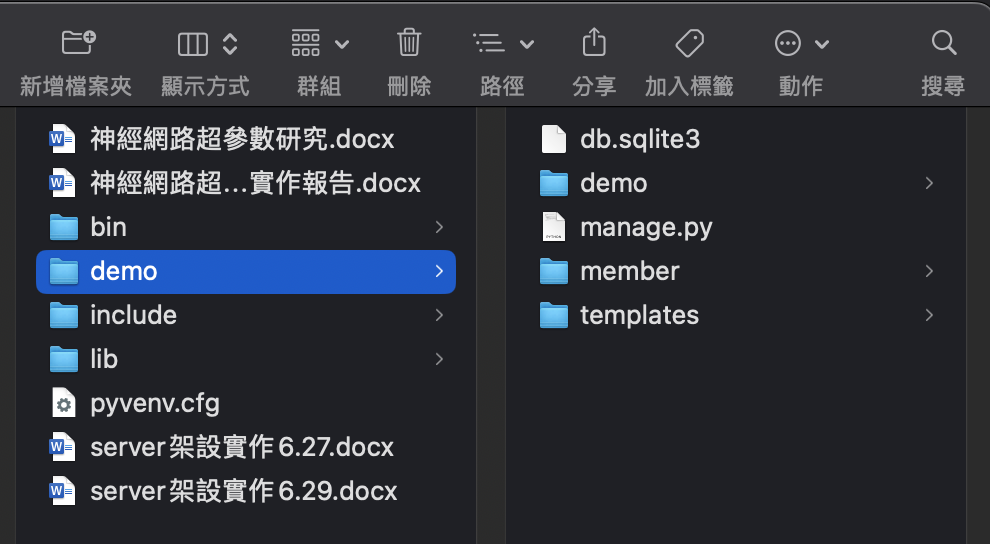


進入原資料夾後輸入：python manage.py runserver，開始試跑伺服器執行，執行成功輸入本地位址的網址會出現，如果想用特定網路位址執行只要在上面那串指令再空格之後打上特定的位址即可

manage.py是Django提供的命令列工具，可以利用他做很多事，我們這次實作就是負責建立app, 請Django幫你根據Model建立資料庫欄位, 紀錄變更, 啟動開發測試網頁伺服器等等，

一張含有 文字, 軟體, 電腦, 網頁 的圖片

自動產生的描述

接下來就到創建app的部分，輸入指令：python manage.py startapp member後就能在原資料夾底下新增一個名為member的新資料夾，也就是app的名稱

每次新增完一個app後，就要到主資料夾中的settings.py檔案內的INSTALL\_APPS中新增上剛剛app的檔名。

settings.py是設定專案資料夾的相關路徑、時區、資安等等套件的地方。INSTALL\_APPS是儲存目前已知的app的位址，當需要執行特定app的時候會到這裡來確認是否有新增檔名進去，假如是沒新增進去的檔名的檔案，就會因為無法找到而執行失敗

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

要建立一個網頁需要用到templates，因為Django中要從當前的app裡面找到templates資料夾需要有這個設定，如果當前的app目錄裡面的名稱沒有templates的資料夾會找不到特定的template html，簡單說templates就是要給使用者看到的畫面。所以這個templates資料夾需要自己新增，然後在templates資料夾裡新增需要的html檔，最後還需要在settings.py裡面的TEMPLATES新增幾段程式碼：Import os和在”DIRS”裡新增os.path.join(BASE\_DIR, “templates”)

DIRS的部分是要讓Django額外搜尋templates的目錄

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

在剛剛新增的templates裡面的html檔裡把需要顯示在網頁上的變數寫上去，再修改一下一些東西，像是網頁名稱之類的

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

依據在html裡面給的變數，讓他儲存相對應的內容，在member資料夾中的view.py裡，先輸入剛剛html檔名後把view.py連接到剛剛的html檔，再把變數要代表的內容填上去

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

urls.py是設定哪個url要對應到哪一隻Python function。

接下來設定輸入指定網址後要顯示的東西，這樣設定完之後輸入網址：http://127.0.0.1:8000/member\_info/ 會顯示來自剛剛member資料夾中的view.py裡面的show\_member\_info所寫的資訊

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

一張含有 螢幕擷取畫面, Rectangle, 陳列, 電腦 的圖片

自動產生的描述

其他沒有設定到的python程式介紹

在demo主資料夾中的有：

* wsgi.py是設定連接Apache後執行專案的虛擬環境路徑在哪裡。

而wsgi是一種規範或稱協議

* asgi是wsgi的繼承者，支持wsgi不支持的一些協議

在member資料夾中的有：

* admin.py是用來管理網站後台用的，但要實踐管理後台的功能還需要做一些設定才可以
* models.py是設定一些網站上可能會用來和使用者互動的小功能