# Sử dụng Virtualenv để tạo môi trường Python riêng biệt

Virtualenv nhằm tạo ra môi trường Python riêng biệt cho mỗi dự án của Python / Django. Mỗi dự án cần danh sách thư viện khác nhau cũng như phiên bản khác nhau, nếu tất cả cùng cài trong hệ thống sẽ dễ dẫn đến nhiều vấn đề phát sinh.

Để sử dụng virtualenv dùng lệnh:

pip install virtualenv

Sau khi cài đặt xong, tạo môi trường riêng cho từng bằng lệnh:

virtualenv my\_env

Đối với mỗi dự án nên tạo 1 môi trường riêng, tất cả thư viện sử dụng khi cài cho project sẽ nằm trong thư mục này.

Chạy dòng lệnh sau để kích hoạt môi trường ảo:

source my\_env/bin/activate

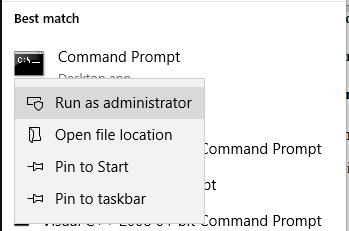
Để bỏ kích hoạt:

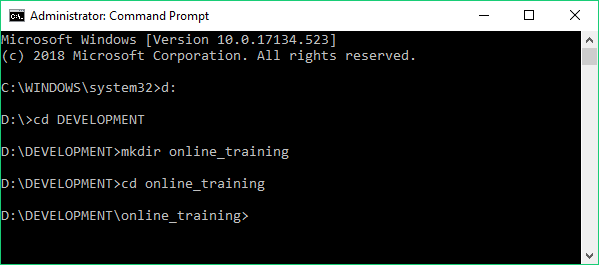
source my\_env/bin/deactivate

Ví dụ cụ thể tạo môi trường ảo cho dự án (giả sử virtualenv đã được cài hoàn thành)

Dự án Online\_Traning trong thư mục D:\Development

**Bước 1: sử dụng cmd (Run as Administrator) tạo thư mục cho dự án**

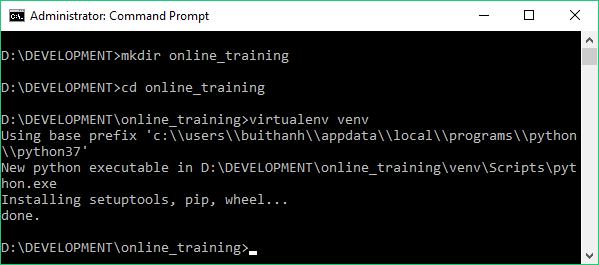
****



**Bước 2: tạo môi trường ảo cho dự án**

Sử dụng lệnh sau:

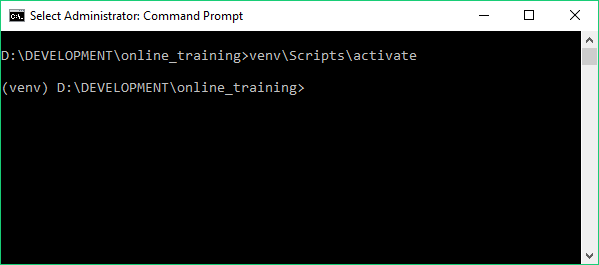
virtualenv venv



**Bước 3: sử dụng môi trường ảo này**

Sử dụng lệnh:

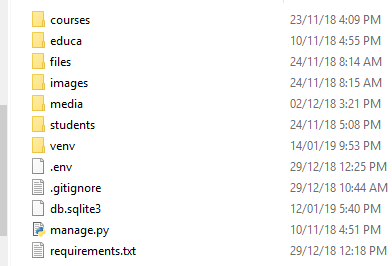
venv\Scripts\activate



(khi sử dụng virtualenv thì sẽ có dòng chữ (venv) phía trước như trong hình)

Lúc này dự án khi cài thư viện sẽ nằm trong thư mục venv này.

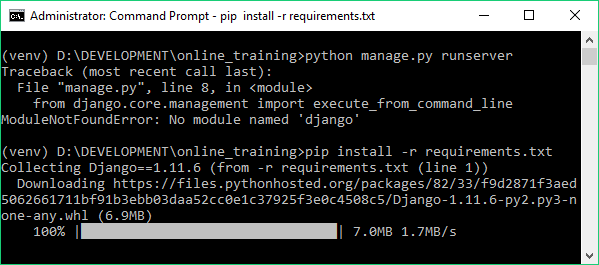
Thư mục dự án lúc này sẽ như sau (giả sử source đã có):



Với requirements.txt chứa các thư viện cần cài đặt.

Cài đặt các thư viện trong file này như sau:

pip install -r requirements.txt



# Các file tự thiết lập cho dự án

## File .env

SECRET\_KEY=số sercret key

DEBUG=True hay False

ALLOWED\_HOSTS=.localhost,127.0.0.1

DATABASE\_URL=mysql://username:password@localhost:3306/tencsdl

## File .gitignore

File này sẽ quy định các file hay thư mục nào không up lên git (vì tính bảo mật hoặc thông tin cấu hình cá nhân)

\_\_pycache\_\_/

\*.py[cod]

.env

venv/

# SQLite database files

\*.sqlite3