**< 프로젝트 뼈대 만들기(Ubuntu) >**

**※ 장고에서 프로젝트와 애플리케이션의 개념**

프로젝트란 개발 대상이 되는 전체 프로그램을 의미하며, 프로젝트를 몇 개의 기능 그룹으로 나누었을 때, 프로젝트 하위의 서브 프로그램을 애플리케이션이라고 말한다. 이런 개념으로 프로젝트 디렉토리와 애플리케이션 디렉토리를 구분하고, 코딩하는 파일도 프로젝트 파일인지 애플리케이션 파일인지 구분해서 적절한 위치에 저장해야 한다.

**※ 다음과 같은 순서로 명령어 실행**

django-admin startproject mysite // mysite라는 프로젝트를 생성함

python manage.py startapp polls // polls라는 애플리케이션을 생성함

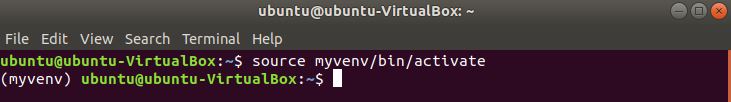
vim settings.py // 설정 파일을 확인 및 수정함

python manage.py migrate // 데이터베이스에 기본 테이블을 생성함

python manage.py runserver // 현재까지 작업을 개발용 웹 서버로 확인함

**※ 모든 작업은 가상환경(virtualenv)안에서 진행됌**

리눅스 --> source myvenv/bin/activate



윈도우 --> myvenv\Scripts\activate

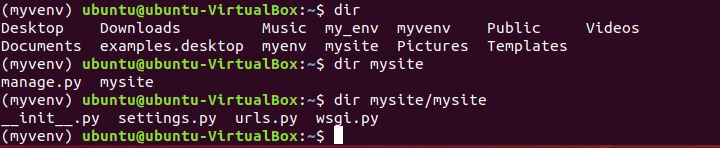
**1. 프로젝트 생성**

$ django-admin startproject mysite

$ dir

$ dir mysite // 상위 디렉토리

$ dir mysite/mysite // 하위 디렉토리

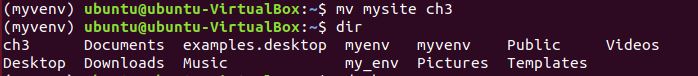


하위 mysite 디렉토리는 프로젝트 디렉토리이고, 상위 mysite 디렉토리는 프로젝트 관련 디렉토리/파일을 모으는 역할만 하는 디렉토리다.

**1.1 혼동을 막기 위해 ch3로 디렉토리명 변경**

$ mv mysite ch3

$ dir

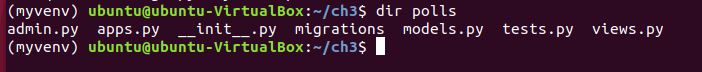


**2. 애플리케이션 생성**

$ cd ch3

$ python manage.py startapp polls

$ dir polls



모든 애플리케이션 개발에 반드시 필요한 파일들은 장고가 알아서 생성해주고, 개발자들은 그 내용을 채워넣기만 하면 된다.

**3. 프로젝트 설정 파일 변경**

프로젝트에 필요한 설정값들은 settings.py 파일에 지정. settings.py 파일은 프로젝트의 전반적인 사항들을 저장해주는 곳으로, 루트 디렉토리는 포함한 각종 디렉토리의 위치, 로그의 형식, 프로젝트에 포함된 애플리케이션의 이름 등이 지정되어 있다.

$ cd ch3/mysite

$ vim settings.py

**3-1. ALLOWED\_HOSTS 항목 지정**

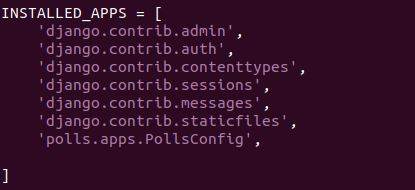
장고는 DEBUG=True이면 개발 모드로, False이면 운영 모드로 인식한다. 운영 모드인 경우는 ALLOWD\_HOSTS에 반드시 서버의 IP나 도메인을 지정해야 하고, 개발 모드인 경우에는 값을 지정하지 않아도 ['localhost', '127.0.0.1']로 간주한다.



**3-2. polls 애플리케이션 등록**

프로젝트에 포함되는 애플리케이션들은 모두 설정 파일에 등록되어야 한다. 애플리케이션을 등록할 때는 간단하게 애플리케이션의 모듈명인 'polls'만 등록해도 되지만, 애플리케이션의 설정 클래스로 등록하는 것이 더 정확한 방법이다.

'polls.apps.PollsConfig' (추가)



**3-3. 데이터베이스 엔진**

장고는 디폴트로 SQLite3 데이터베이스 엔진을 사용하도록 설정되어 있다. 다른 데이터 엔진으로 변경할 수도 있으며, 만약 MySQL이나 Oracle, PostgreSQL 등 다른 데이터베이스로 변경하고 싶다면 settings.py 파일에서 수정해주면 된다.



**3-4. 타임존 지정**

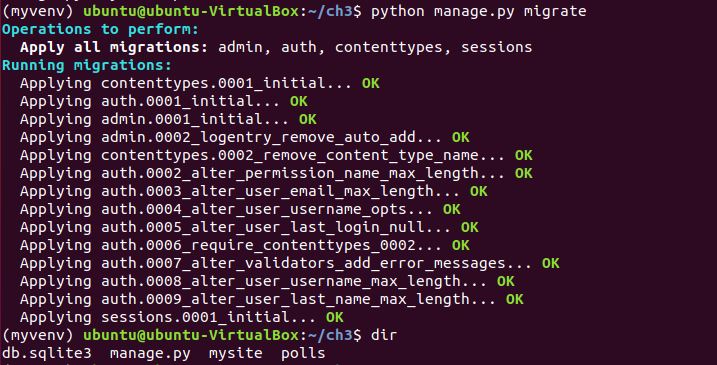
최초에는 세계표준시(UTC)로 되어 있는데, 한국 시간으로 변경한다.



**4. 기본 테이블 생성**

$ cd ch3 // $ cd ..

$ python manage.py migrate



migrate는 데이터베이스에 변경사항이 있을 때 이를 반영해주는 명령어다.

실행 결과로는 SQLite3 데이터 베이스 파일인 db.sqlite3 파일이 생성된다.

**5. 지금까지 작업 확인하기**

장고에서는 개발 과정 도중에 현재 상태를 확인해 볼 수 있도록 runserver라고 하는 테스트용 웹 서버를 제공해준다.

$ cd ch3

$ python manage.py runserver 0.0.0.0:8000

명령 입력 시 자신의 서버에 맞는 IP 주소와 포트번호를 입력하면 된다.

**※ runserver 실행 방법**

python manage.py runserver

-> IP 주소와 포트번호를 지정하지 않으면, 디폴트로 127.0.0.1 주소 및 8000번 포트를 사용한다.

python manage.py runserver 8888

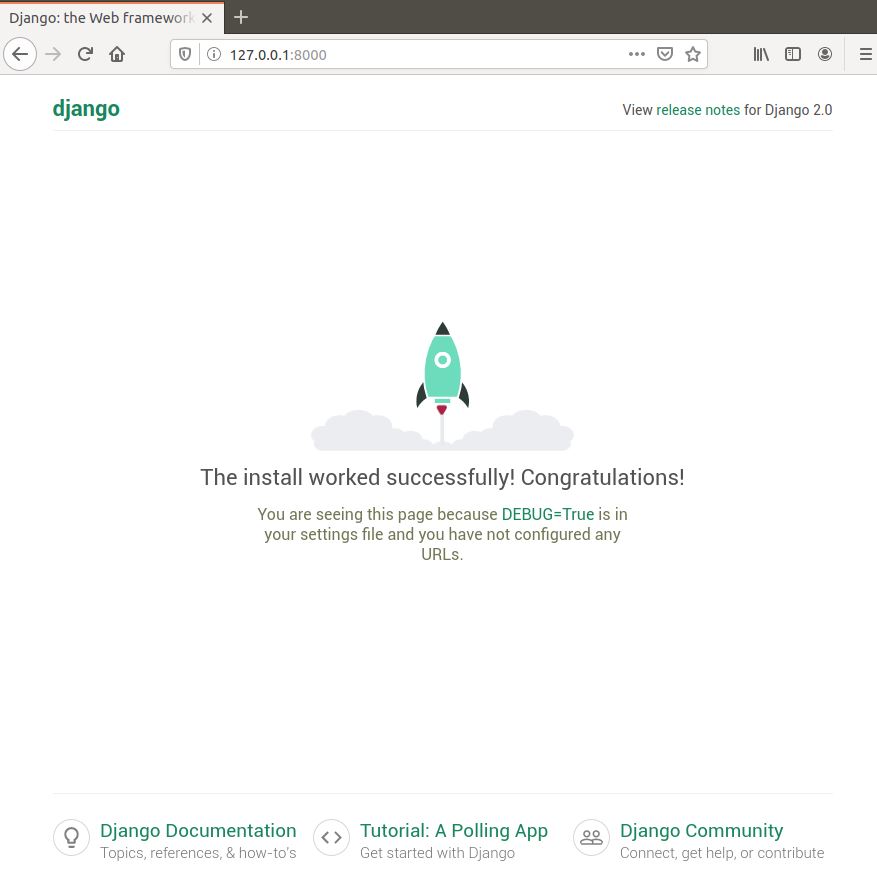
-> 포트번호만 지정하면, 디폴트 127.0.0.1 주소 및 지정한 8888번 포트를 사용한다.

python manage.py runserver 0.0.0.0:8000 &

&를 명령어 끝에 추가하면, 웹 서버 프로그램이 백그라운드에서 실행된다. 단 리눅스와 맥에서만 가능하다.

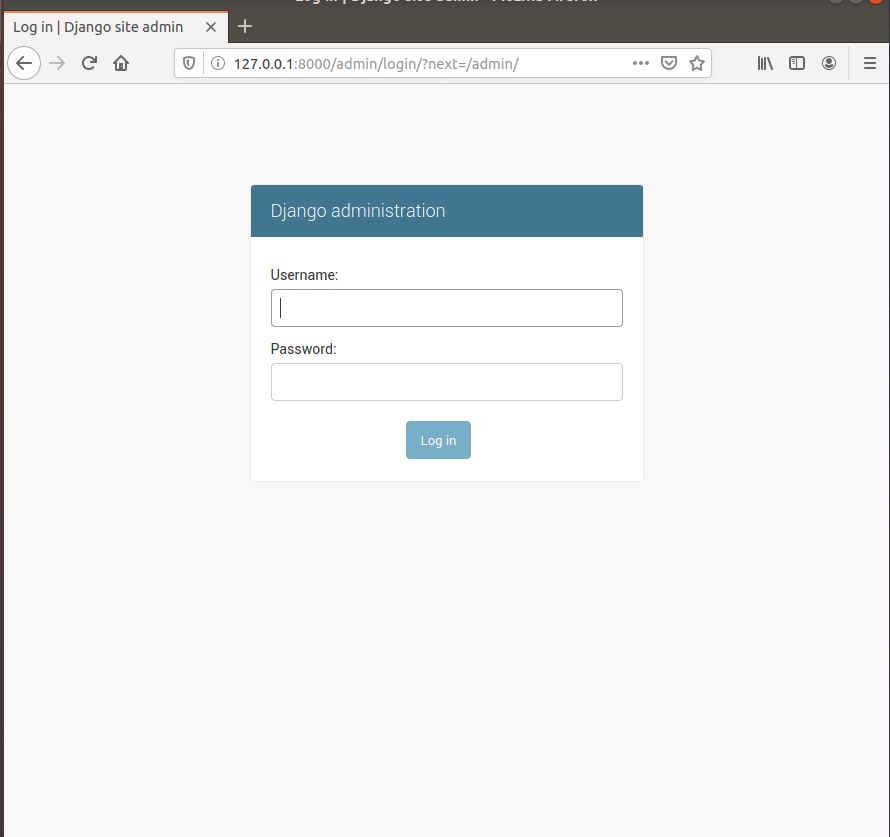
**- runserver가 정상적으로 실행되면 웹 브라우저에서 기존에 설정한 IP 주소로 접속한다.**

http://127.0.0.1:8000



**- 장고에서 기본적으로 제공하는 Admin 사이트에 접속하여 테이블이 생성된 것을 확인한다.**

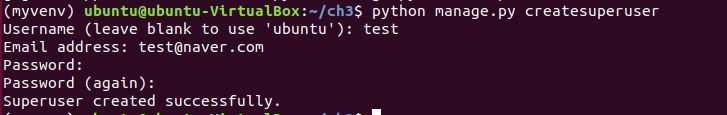
http://127.0.0.1:8000/admin



**- Admin 사이트에 로그인 하기 위해서는 관리자(슈퍼유저)를 만들어야 한다.**

$ cd ch3

$ python manage.py createsuperuser



**- 관리자 생성후 로그인하면 다음과 같은 페이지가 나타난다.**

