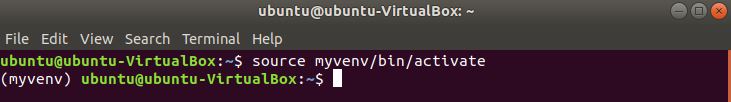
**< Django의 핵심 기능 (Ubuntu)>**

**※ 작업은 가상환경(virtualenv)안에서 진행됌**

리눅스 --> source myvenv/bin/activate



윈도우 --> myvenv\Scripts\activate

**1. Admin 사이트 꾸미기**

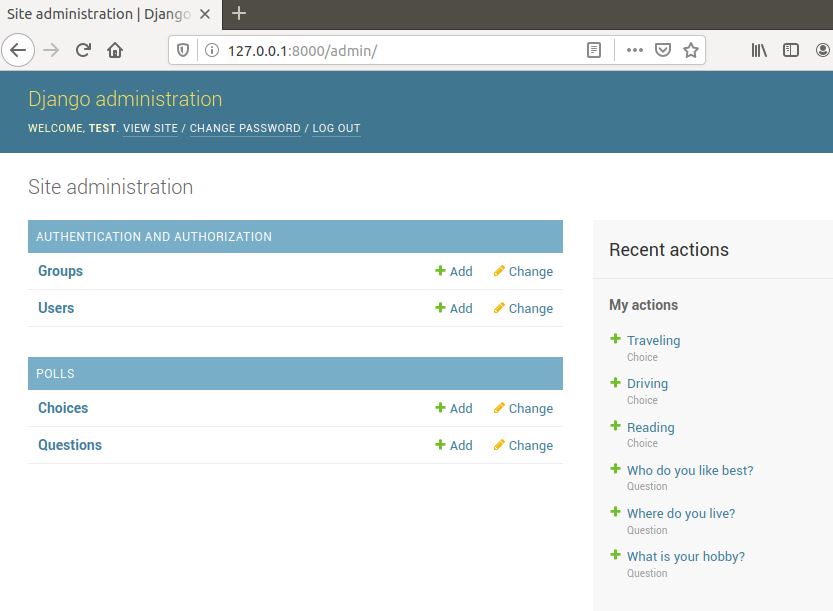
runserver 실행

cd ~/ch3에서 확인

$ python manage.py runserver 0.0.0.0:8000

**1-1. 데이터 입력 및 수정**

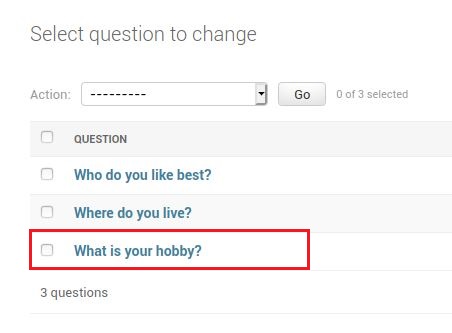
**- Admin 사이트의 첫 화면**



정상적으로 로그인 되면 위 그림처럼 Admin 사이트의 첫 화면이 나타나고, 여기엔 장고에서 제공해주는 Users, Groups 정보와 일전에 만든 Question, Choices 정보가 보인다.

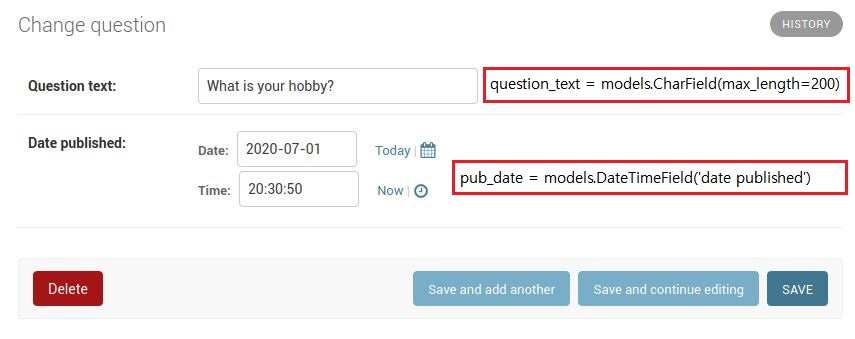
첫 화면에서 보이는 [Add] 버튼 또는 [Change] 버튼을 클릭하거나, [Questions]라는 제목을 클릭하면 아래 그림과 같은 테이블 상세 정보 페이지가 나타나고 여기서 테이블 입력, 수정을 할 수 있다.

**- Questions 테이블 상세 정보 페이지**



원하는 질문을 클릭하면 다음 그림과 같이 선택한 레코드의 상세 페이지가 나타난다. 이 화면은 models.py에 정의한 테이블 모델을 기초로 장고가 자동으로 생성해주는 것이다.

**- 레코드 상세정보**



'레코드 상세정보' 에서 각 필드의 값, 즉 테이블의 컬럼 값을 변경할 수 있다. Question text 필드나 Date, Time 필드 값을 임의로 수정해보고, 변경 내용을 테이블에 반영하기 위해 저장 버튼을 클릭해주면 된다.

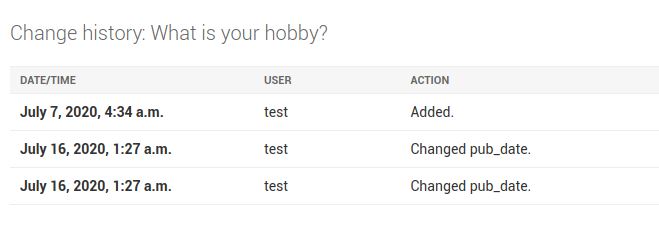
· Save : 변경사항을 저장하고 현 객체의 리스트를 보여주는 페이지로 돌아간다.

· Save and continue editing : 변경사항을 저장하고 현재의 페이지를 다시 보여준다.

· Save and add another : 변경사항을 저장하고 새로운 레코드를 추가할 수 있는 폼 페이지를 보여준다.

· Delete : 삭제 확인 페이지를 보여준다.

**- 변경 이력\_History 화면**



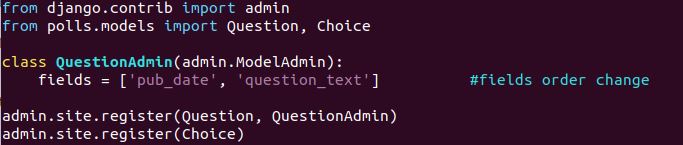
**1-2. 필드 순서 변경하기**

테이블의 데이터를 변경하는 것이 아니라, 테이블을 보여주는 UI 양식을 변경하려면 polls/admin.py 파일을 변경하면 된다.

$ cd ~ch3/polls

$ vim admin.py

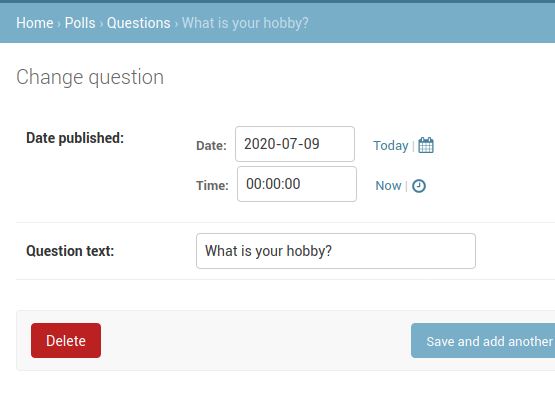
**- QuestionAdmin 클래스 정의 -> admin.py**



ModelAdmin 클래스를 상속받아 새로운 QuestionAdmin 클래스를 정의하고,

그 클래스를 admin.site.register() 함수의 두 번째 인자로 등록해주면 된다.

**- QuestionAdmin 클래스 정의 -> 필드 순서 변경**



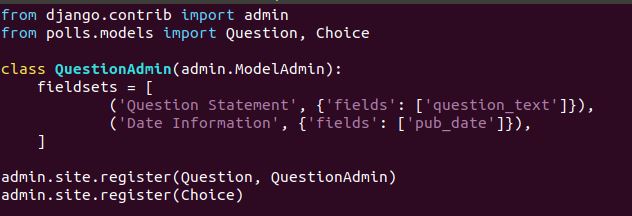
**1-3. 각 필드를 분리해서 보여주기**

polls/admin.py 파일 수정

$ cd ~ch3/polls

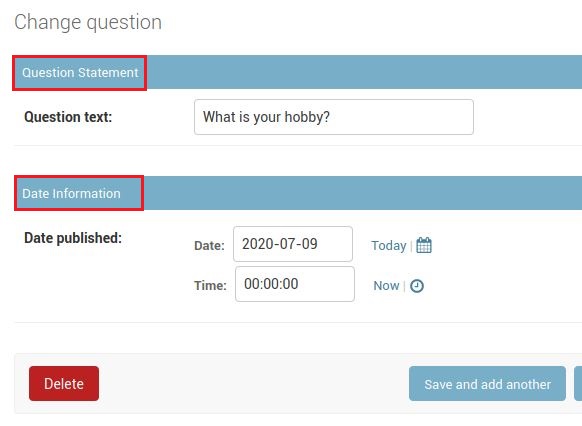
$ vim admin.py

**- 필드를 분리해서 보여주기 -> polls/admin.py 코딩**



fieldsets에 있는 각 튜플의 첫 번째 인자가 해당 필드의 제목이 된다.

**- 필드를 분리해서 보여주기 - UI 확인**

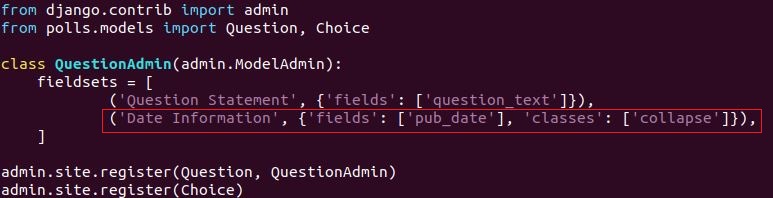


**1-4. 필드 접기**

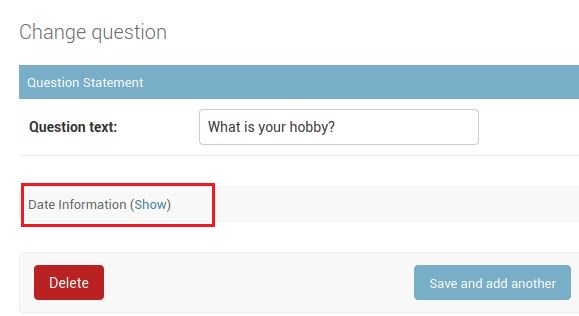
필드 순서 변경(1-2), 필드 분리(1-3) 및 필드 접기(1-4) 등은 필드 개수가 많아 폼이 길어진 경우 유용하게 사용할 수 있다.

$ vim admin.py (기존 polls/admin.py 소스에서 한 줄만 변경하면 됌)

**- 필드를 접어서 보여주기 -> polls/admin.py 코딩**



**- 필드를 접어서 보여주기 -> UI 확인**

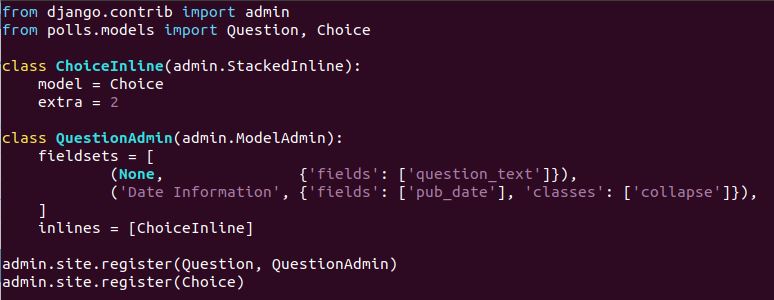


**1-5. Question 및 Choice를 한 화면에서 변경하기**

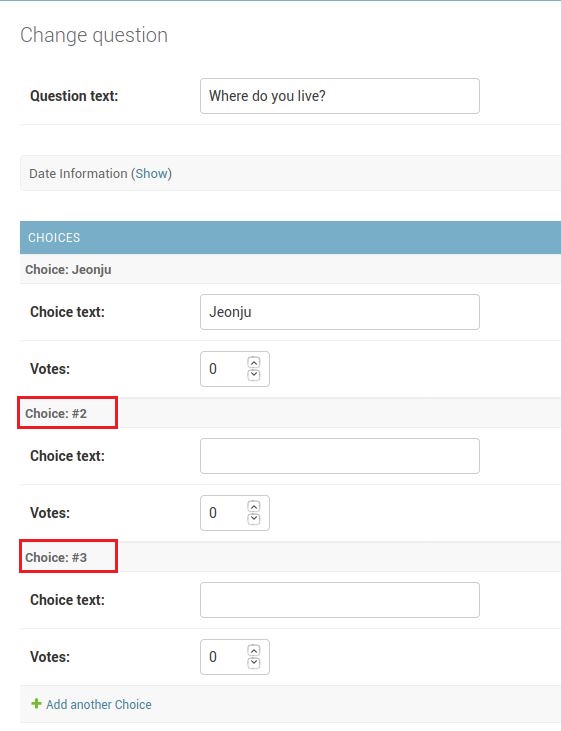
아래의 polls/admin.py 파일을 수정하면 Question과 Choice text를 같이 보면서 수정 또는 추가할 수 있다. Question 레코드들 기준으로 여러개의 Choice 레코드가 연결되는 것이므로, ChoiceAdmin이 아니라 QuestionAdmin 클래스를 수정한다.

$ vim admin.py

**- Queston과 Choice 같이 보기 -> polls/amdin.py 코딩**



**- Question과 Choice 같이 보기 -> StackedInline**

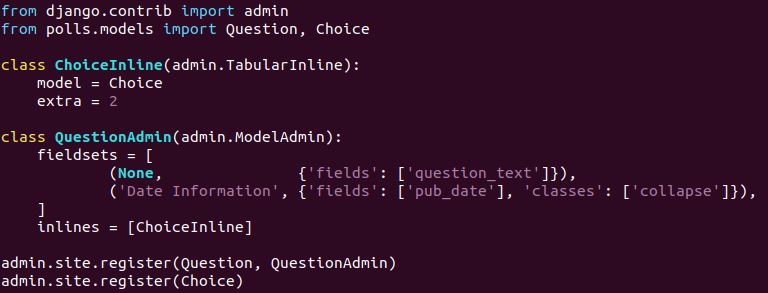


소스코드에서 extra 변수로 지정한 값에 따라, 한 번에 보여지는 Choice text의 숫자가 결정된다.

**1-6. 테이블 형식으로 보여주기**

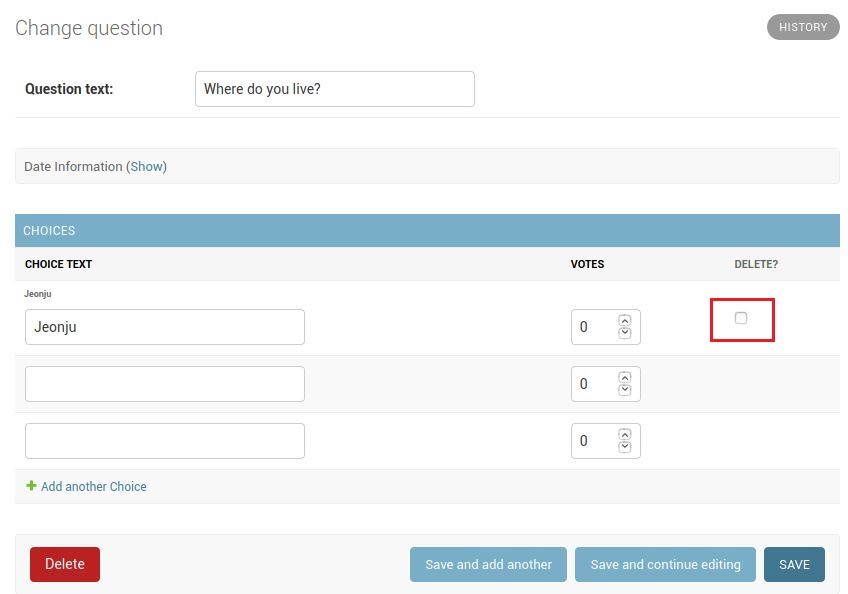
$ vim admin.py (기존 polls/admin.py 소스에서 한 줄만 변경하면 됌)

**- 테이블 형식으로 보여주기 -> polls/admin.py 코딩**



소스 코드를 변경하면 아래 그림처럼 테이블 형식으로 보여준다. 우측의 [Delete] 컬럼에 체크 표시를 하고 [SAVE] 버튼을 클릭하면, 해당 답변 항목을 삭제할 수 있다.

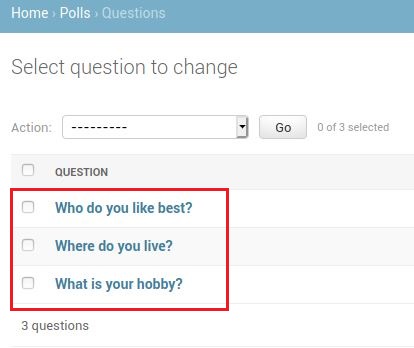
**- 테이블 형식으로 보여주기 -> TabularInline**



**1-7. 레코드 리스트 컬럼 지정하기**

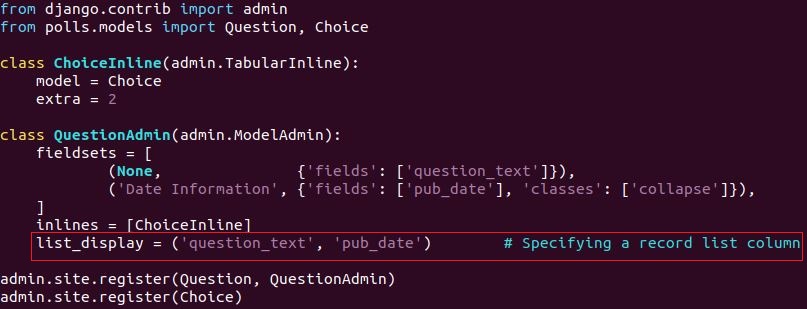
Admin 사이트의 첫 페이지에서 테이블명을 클릭하면, 해당 테이블의 레코드 리스트가 나타난다. 장고는 디폴트로 models.py에서 정의한 \_\_str\_\_() 메소드의 리턴값을 레코드의 제목으로 사용한다.

**- Question 레코드 리스트를 디폴트 형식으로 보여주기**



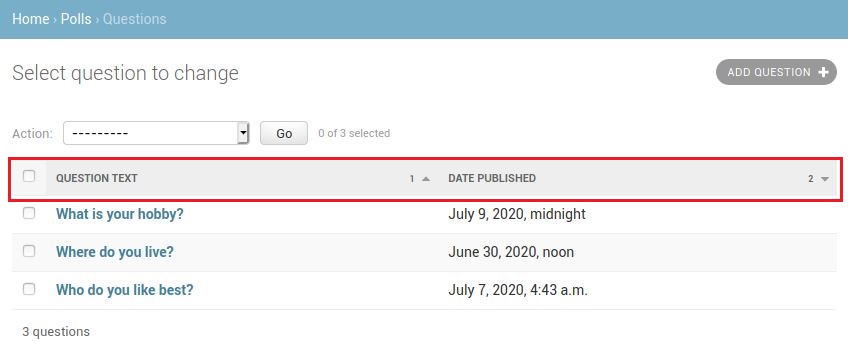
(디폴트 형식으로 바꾸기) polls/admin.py 파일에 list\_display 속성을 한 줄 추가하면, 레코드 리스트에 보여주는 컬럼 항목을 지정 할 수 있다.

**- 레코드 리스트 컬럼 지정하기 -> polls/admin.py 코딩**



아래 그림을 보면 제목이 변경되었고, DATE PUBLSHED 컬럼이 추가되었다. 각 컬럼의 헤더를 클릭하면 해당 컬럼을 기준으로 정렬하여 보여준다.

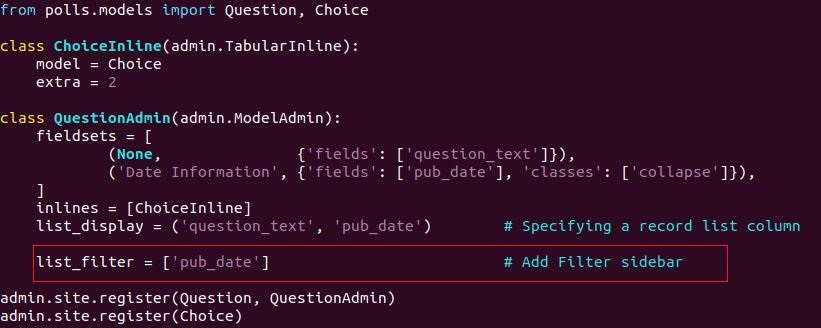
**- 레코드 리스트 항목 지정하기 -> UI 확인**



**1-8. list\_filter 필터**

polls/admin.py 파일에 list\_filter 속성을 한 줄 추가하면, UI 화면 우측에 필터 사이드 바를 붙일 수 있다.

**- list\_filter 필터 추가 -> polls/admin.py 코딩**



[FILTER]에는 필터에 사용된 기준 필드 타입에 따라, 장고가 자동으로 적절한 항목들을 보여준다.

아래 그림에서 pub\_date 필드의 타입이 DateTimeField 타입이므로, 장고가 이에 따라, Any date, Today 등의 옵션 항목을 제공해주고 있다.

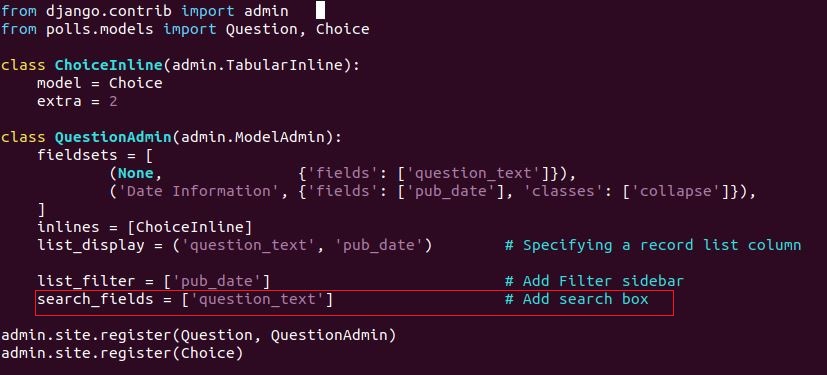
**- list\_filter 필터 추가 -> UI 확인**



**1-9. search\_fields**

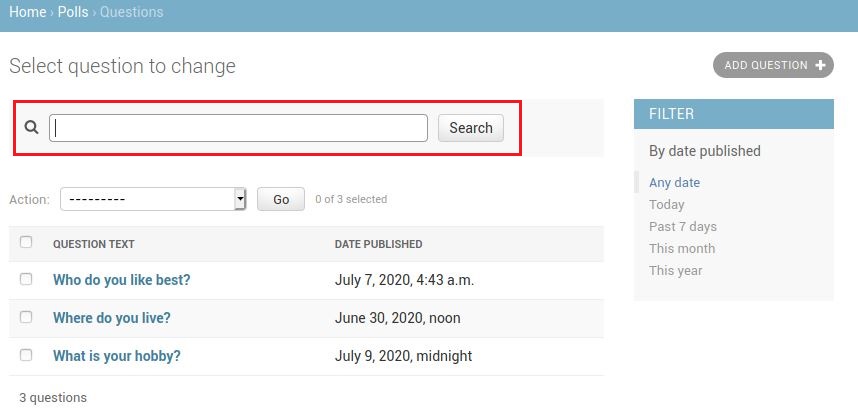
polls/admin.py 파일에 search\_fields 속성을 한 줄 추가하면, UI 화면에 검색 박스를 표시할 수 있다.

**- search\_fields 검색 박스 추가 -> polls/admin.py**



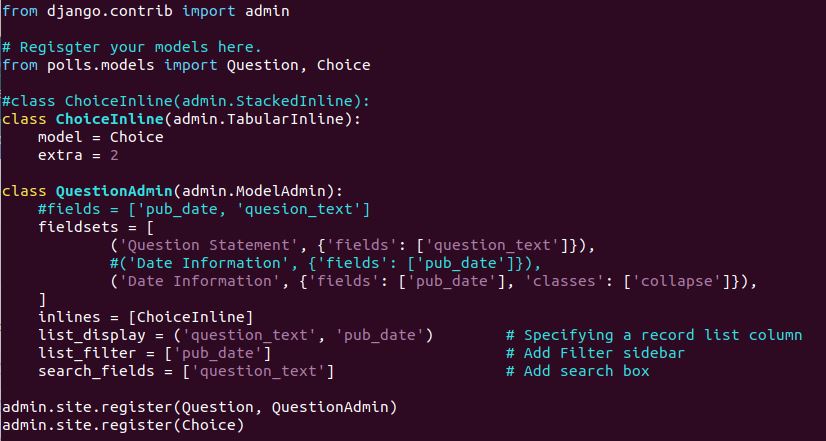
검색 박스에 단어를 입력하면 장고는 LIKE 쿼리로 question\_text 필드를 검색한다. search\_fields 속성에 여러 개의 필드를 지정하면 지정한 모든 필드에서 입력된 단어를 검색한다.

**- search\_fields 검색 박스 추가 -> UI 확인**



**1-10. polls/admin.py 변경 내용**

- polls/admin.py 최종 모습



**1-11. Admin 사이트 템플릿 수정**

ch3 디렉토리에서 진행

$ mkdir templates

$ mkdir templates/admin

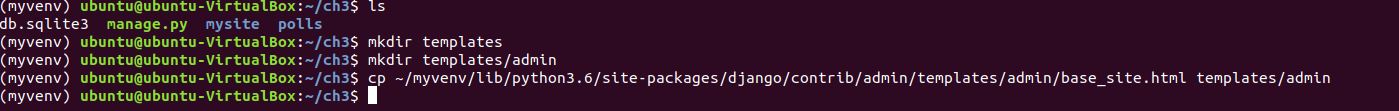
$ cp ~/myvenv/lib/python3.6/site-packages/django/contrib/admin/template/admin

/base\_site.html templates/admin

(tab 없음, cp ~/myvenv 부분에서 tempaltes/admin 까지 한줄로 작성)

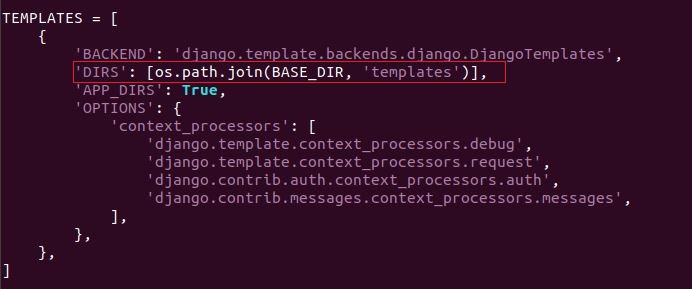
\* 장고 설치 디렉토리 확인 명령어(위 경로 확인 시)

$ python -c "import django; pirnt(django.\_\_path\_\_)"



$ vim mysite/settings.py

**- base\_site.html 복사 및 settings.py 수정**

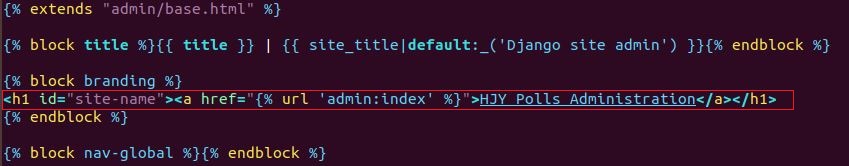


복사한 base\_site.html 파일을 원하는 내용으로 수정

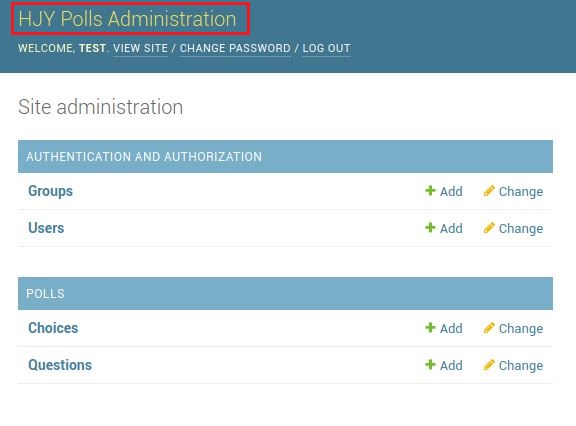
$ cd ~/ch3/templates/admin

$ vim base\_site.html

**- base\_site.html 수정**



**- base\_site.html 수정 -> UI 확인**



이와 같은 방식으로, 장고에서 기본적으로 제공하는 템플릿 파일을 장고의 설치 디렉토리에서 사용자의 프로젝트 디렉토리로 복사해온 다음, 그 파일을 수정하면 사용자의 취향에 맞게 Admin 사이트의 룩앤필을 변경할 수 있다.

한 가지 유의 사항은 템플릿 파일을 복사해올 때 타깃 디렉토리를 templates가 아니라 templates/admin 디렉토리로 해야 장고가 템플릿 파일을 찾을 수 있다.