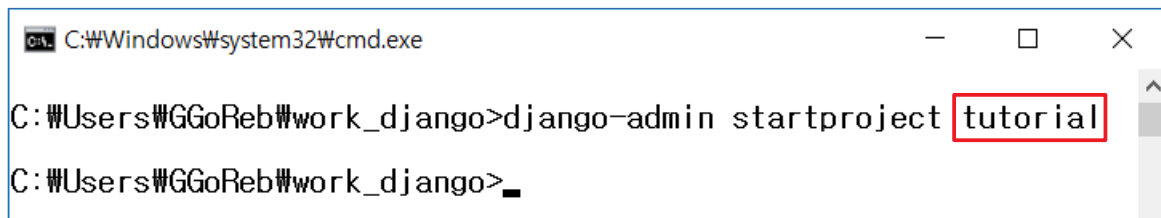


■ 프로젝트 생성 절차

● 프로젝트 기본 구조 생성

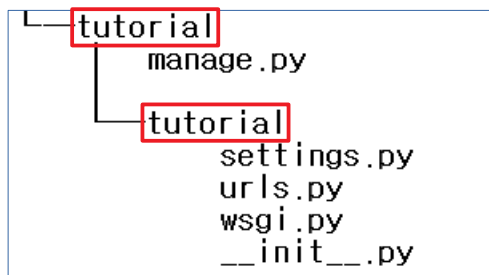
– django-admin startproject [프로젝트명]



```
C:\Windows\system32\cmd.exe

C:\Users\WGoReb\work_django>django-admin startproject tutorial
C:\Users\WGoReb\work_django>
```

– 생성 결과



■ 프로젝트 생성 절차

● App 생성

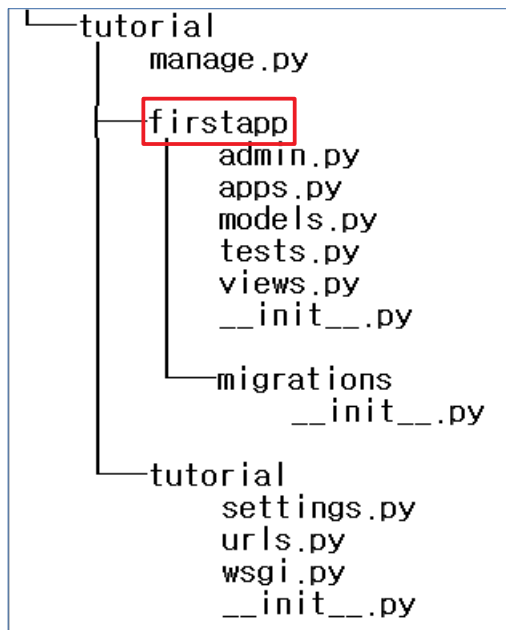
– django-admin startapp [App명]

```
C:\Windows\system32\cmd.exe

C:\Users\WGoReb\work_django>cd tutorial
C:\Users\WGoReb\work_django\tutorial>django-admin startapp firstapp
C:\Users\WGoReb\work_django\tutorial>
```

manage.py 사용 가능

– 생성 결과



■ 프로젝트 생성 절차

● 프로젝트 실행

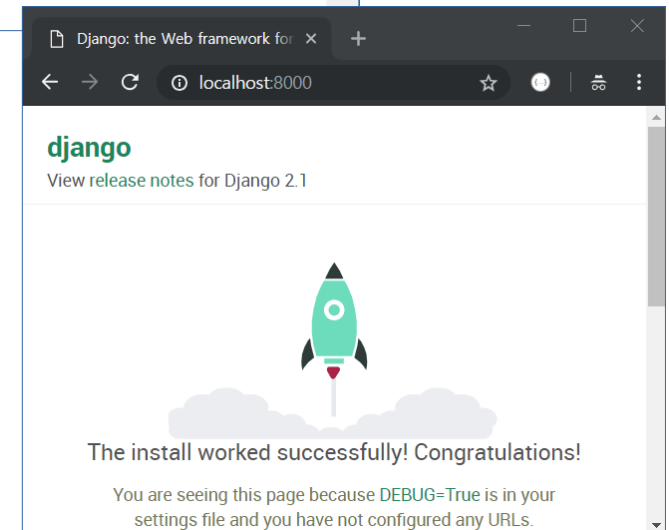
– manage.py runserver [port]

```
C:\Windows\system32\cmd.exe - manage.py runserver

C:\Users\WGoReb\work_django\tutorial>manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
September 15, 2018 - 20:20:57
Django version 2.1.1, using settings 'tutorial.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

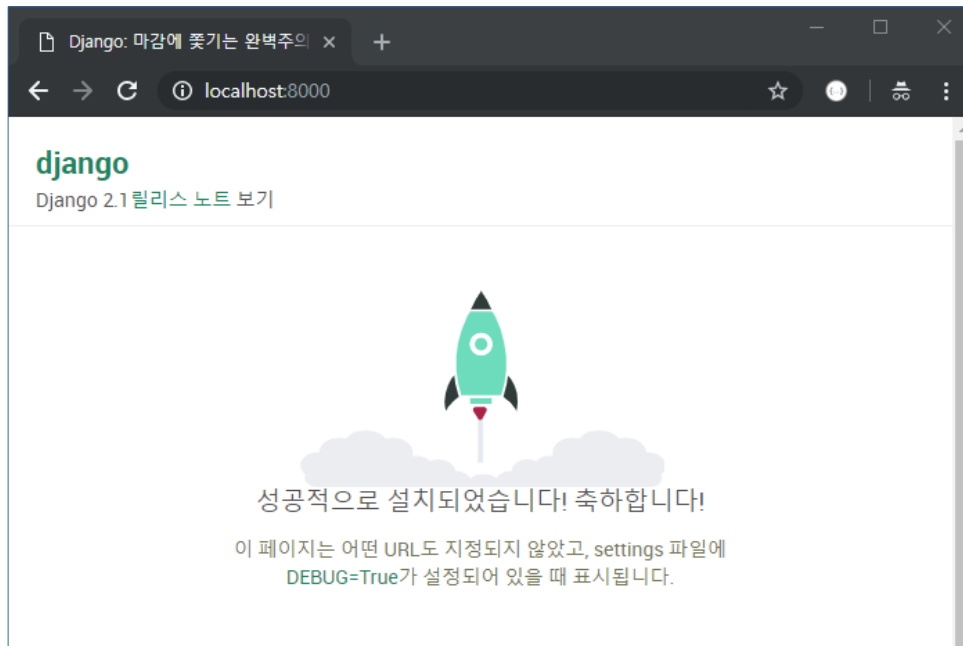


■ 프로젝트 생성 절차

● 언어 및 시간 설정 변경

– settings.py

```
LANGUAGE_CODE = 'ko-kr'  
# LANGUAGE_CODE = 'en-us'  
  
TIME_ZONE = 'Asia/Seoul'  
# TIME_ZONE = 'UTC'
```



■ 브라우저로 접속할 수 있는 웹페이지(주소) 생성 - 1

● 보여질 내용

– firstapp/views.py

```
from django.shortcuts import render
from django.http import HttpResponse

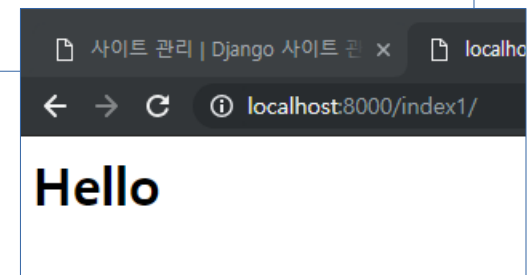
def index1(request):
    return HttpResponse('<h1>Hello</h1>')
```

● 주소 생성 및 내용 연결

– tutorial/urls.py

```
from django.contrib import admin
from django.urls import path
from firstapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index1/', views.index1),
]
```



■ 브라우저로 접속할 수 있는 웹페이지(주소) 생성 - 1

● 보여질 내용

– firstapp/views.py

```
from django.shortcuts import render
from django.http import HttpResponse

def index1(request):
    return HttpResponse('<h1>Hello</h1>')

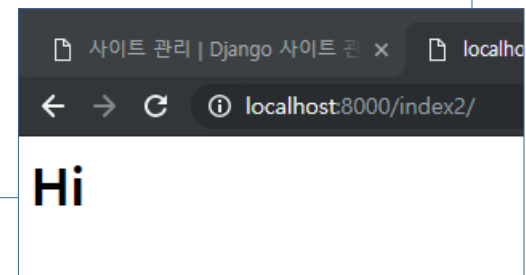
def index2(request):
    return HttpResponse('<h1>Hi</h1>')
```

● 주소 생성 및 내용 연결

– tutorial/urls.py

```
from django.contrib import admin
from django.urls import path
from firstapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index1/', views.index1),
    path('index2/', views.index2),
]
```



■ 브라우저로 접속할 수 있는 웹페이지(주소) 생성 - 2

● 보여질 내용

– firstapp/views.py

```
from django.shortcuts import render
from django.http import HttpResponse

def index1(request):
    return HttpResponse('<h1>Hello</h1>')

def index2(request):
    return HttpResponse('<h1>Hi</h1>')

def main(request):
    return HttpResponse('<h1>Main</h1>')
```

● 주소 생성 및 내용 연결 - app

– firstapp/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('main/', views.main, name='main'),
]
```

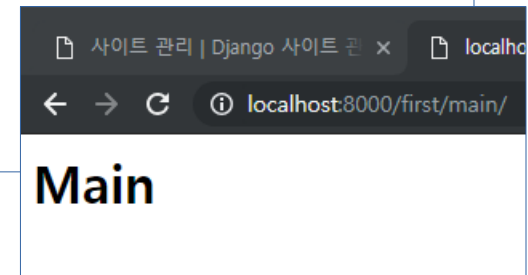
■ 브라우저로 접속할 수 있는 웹페이지(주소) 생성 - 2

● 주소 생성 및 내용 연결 - project

- tutorial/urls.py

```
from django.contrib import admin
from django.urls import path, include
from firstapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index1/', views.index1),
    path('index2/', views.index2),
    path('first/', include('firstapp.urls')),
]
```



■ 테이블 및 속성 정의

● 모델

– firstapp/models.py

```
from django.db import models

class Curriculum(models.Model):
    name = models.CharField(max_length=255)
```

curriculum	
id	name
1	<u>linux</u>
2	python
3	<u>html/css/js</u>
4	<u>django</u>

● 프로젝트에 사용될 APP 등록

– settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'firstapp', # firstapp.apps.FirstappConfig
]
```

■ 테이블 및 속성 정의

● 모델 내용 반영

– python manage.py makemigrations

```
Terminal
+ C:\Users\GGoreb\work_django\tutorial>python manage.py makemigrations firstapp
X Migrations for 'firstapp':
  firstapp\migrations\0001_initial.py
  - Create model Curriculum
```

– python manage.py migrate

```
Terminal
+ C:\Users\GGoreb\work_django\tutorial>python manage.py migrate
X Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
```

■ 테이블 및 속성 정의

● 모델 내용 반영

▼	테이블 (12)
>	auth_group
>	auth_group_permissions
>	auth_permission
>	auth_user
>	auth_user_groups
>	auth_user_user_permissions
>	django_admin_log
>	django_content_type
>	django_migrations
>	django_session
▼	firstapp_curriculum
	id integer
	name varchar(255)
>	sqlite_sequence

※ 데이터베이스 작업 실패시

1. 데이터베이스 관련 파일 모두 삭제 후 재시도
2. 관련 파일만 삭제 (기존 데이터 유지)
 - python manage.py makemigrations
 - python manage.py showmigrations
 - python manage.py migrate --fake 앱 zero
 - 마이그레이션 파일 삭제 0001_initial.py 등
 - python manage.py makemigrations
 - python manage.py migrate --fake-initial

■ 데이터 조회

● 보여질 내용

– firstapp/views.py

```
def show(request):  
    curriculum = Curriculum.objects.all()  
    html = ''  
    for c in curriculum:  
        html += c.name + '<br>'  
    return HttpResponse(html)
```

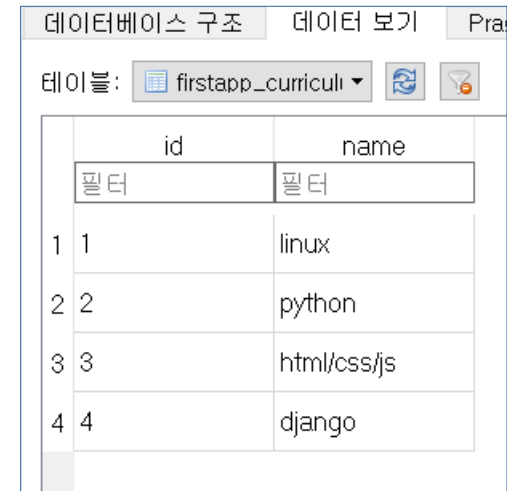
● 주소 생성 및 내용 연결

– firstapp/urls.py

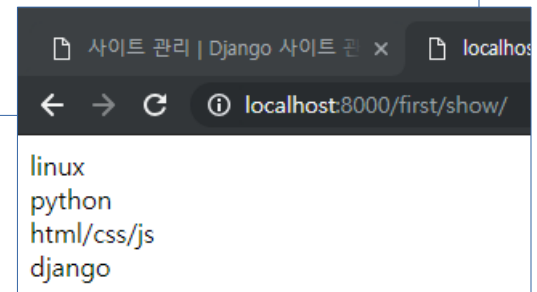
```
urlpatterns = [  
    path('main/', views.main, name='main'),  
    path('show/', views.show, name='show'),  
]
```

● 데이터 입력

– DB tool 이용



	id	name
	<input type="text" value="필터"/>	<input type="text" value="필터"/>
1	1	linux
2	2	python
3	3	html/css/js
4	4	django



■ 관리자 기능 활용

● 관리자 정보 등록

Terminal

+ C:\#Users\#GGoReb\work_django\tutorial> python manage.py createsuperuser

× 사용자 이름 (leave blank to use 'ggoreb'): admin

이메일 주소: admin@ggoreb.com

Password:

Password (again):

비밀번호가 이메일 주소와 너무 유사합니다.

비밀번호가 너무 짧습니다. 최소 8 문자를 포함해야 합니다.

비밀번호가 너무 일상적인 단어입니다.

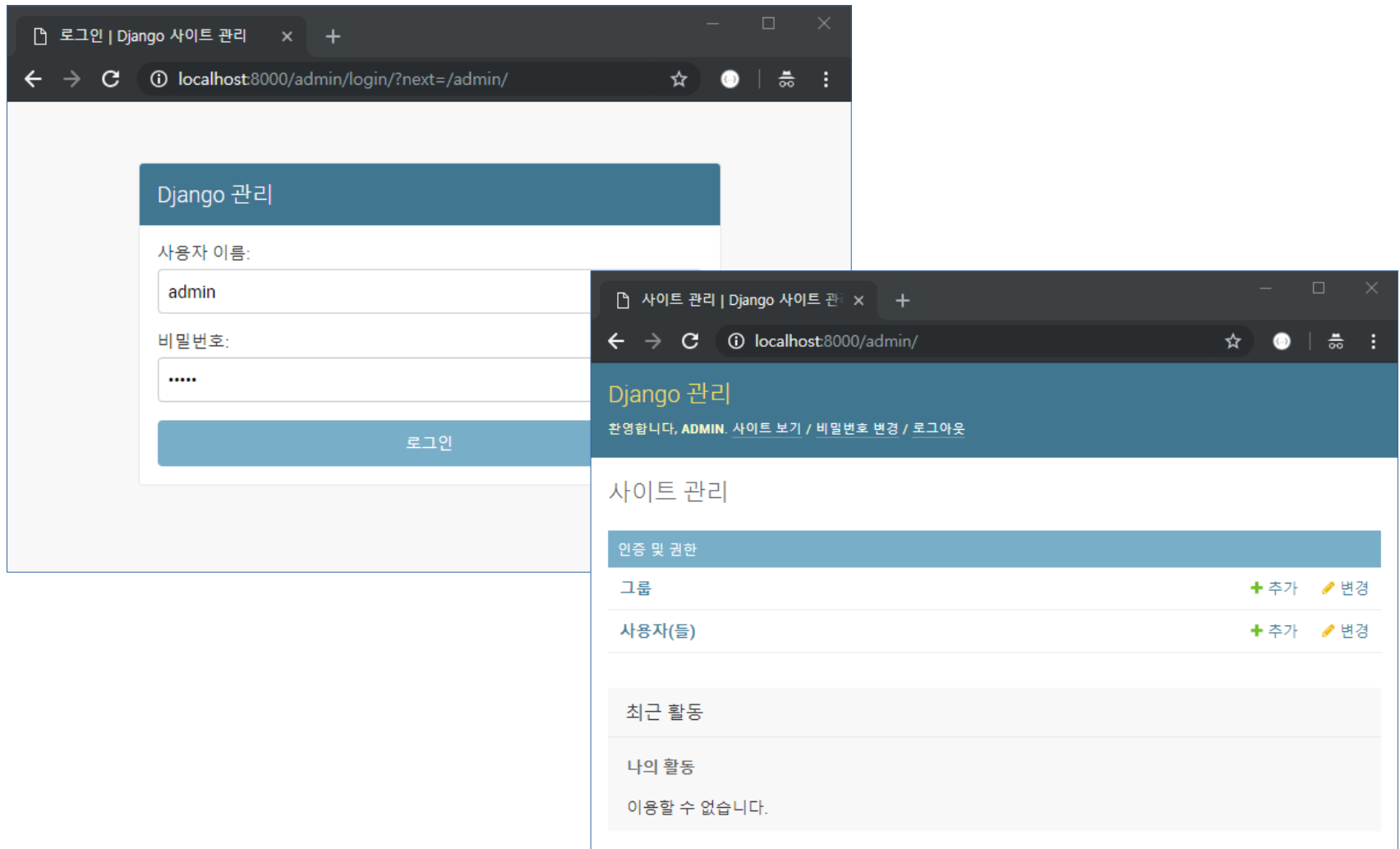
Bypass password validation and create user anyway? [y/N]: y

Superuser created successfully.

■ 관리자 기능 활용

● 관리자 사이트 화면

– `http://localhost:8000/admin`



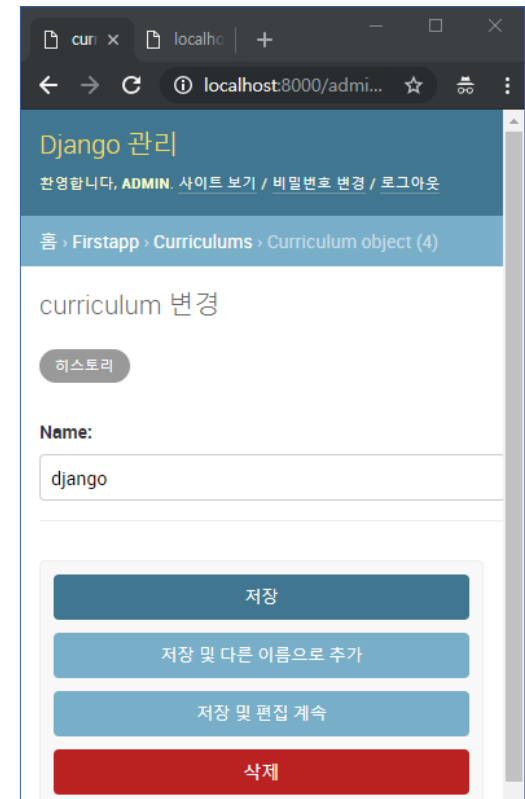
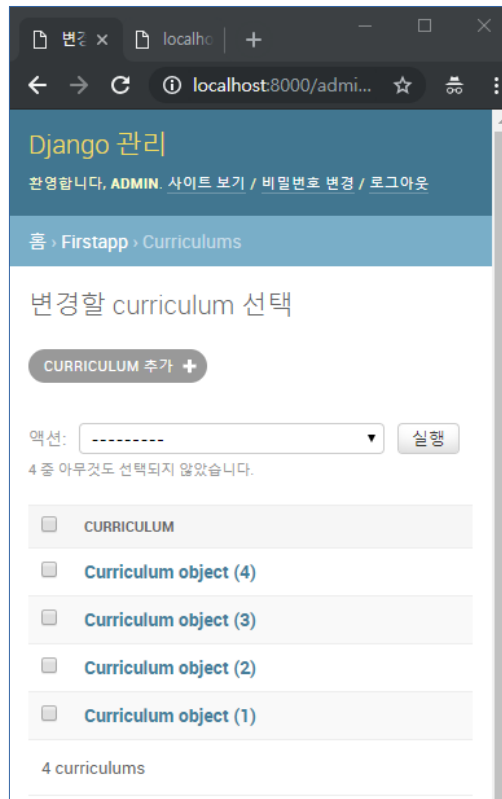
■ 관리자 기능 활용

● 관리자 사이트에서 데이터를 제어 할 수 있도록 등록

– firstapp/admin.py

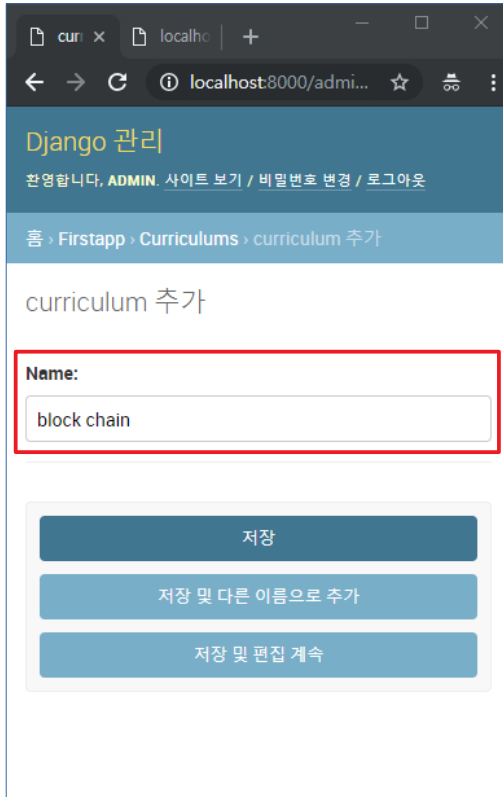
```
from django.contrib import admin
from .models import Curriculum

admin.site.register(Curriculum)
```



■ 관리자 기능 활용

● 관리자 사이트에서 데이터 등록



Django 관리자

환영합니다, ADMIN. [사이트 보기](#) / [비밀번호 변경](#) / [로그아웃](#)

홈 > Firstapp > Curriculums > curriculum 추가

curriculum 추가

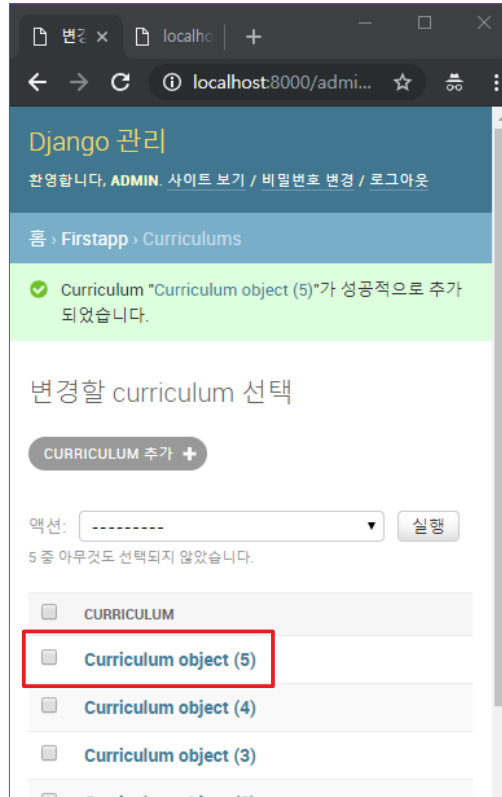
Name:

block chain

저장

저장 및 다른 이름으로 추가

저장 및 편집 계속



Django 관리자

환영합니다, ADMIN. [사이트 보기](#) / [비밀번호 변경](#) / [로그아웃](#)

홈 > Firstapp > Curriculums

✓ Curriculum "Curriculum object (5)"가 성공적으로 추가되었습니다.

변경할 curriculum 선택

CURRICULUM 추가 +

액션: ----- 실행

5 중 아무것도 선택되지 않았습니다.

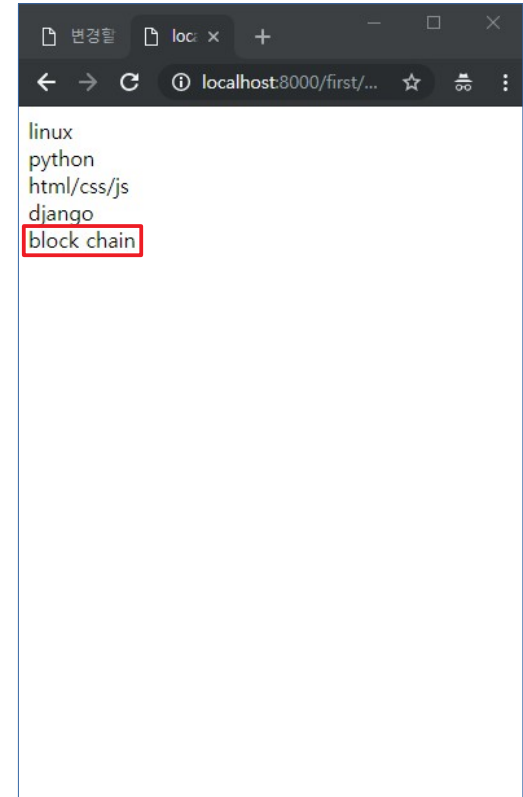
☐ CURRICULUM

☒ Curriculum object (5)

☐ Curriculum object (4)

☐ Curriculum object (3)

☐ Curriculum object (2)



linux

python

html/css/js

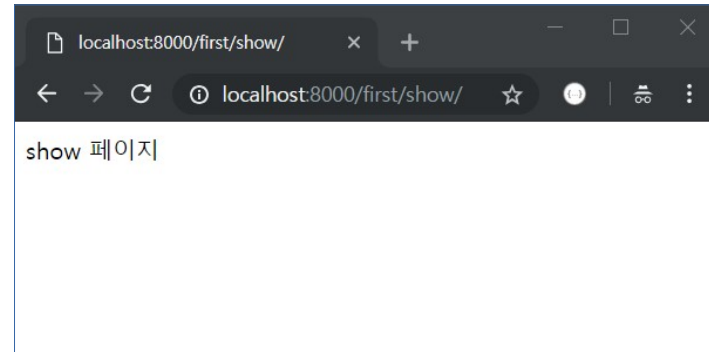
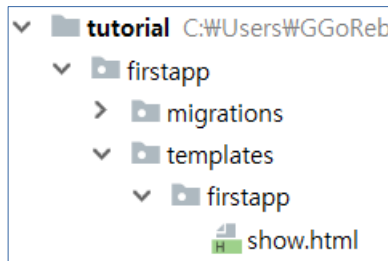
django

block chain

■ 템플릿 활용

● firstapp/templates/firstapp/show.html 생성

– 디렉토리 및 파일 구조



– show.html

```
<body>
  show 페이지
</body>
```

– firstapp/views.py

```
from django.shortcuts import render

def show(request):
    return render(request, 'firstapp/show.html', {})
```

■ 템플릿 활용

● 템플릿 html 데이터 전달

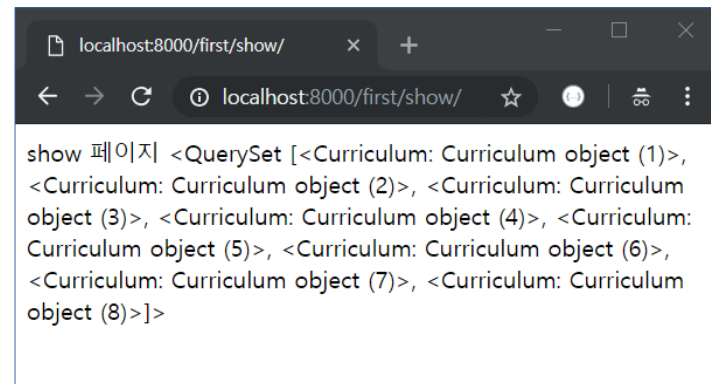
– firstapp/views.py

```
from django.shortcuts import render
from .models import Curriculum

def show(request):
    curriculum = Curriculum.objects.all()
    return render(request, 'firstapp/show.html', {'list': curriculum})
```

– show.html

```
<body>
    show 페이지
    {{ list }}
</body>
```

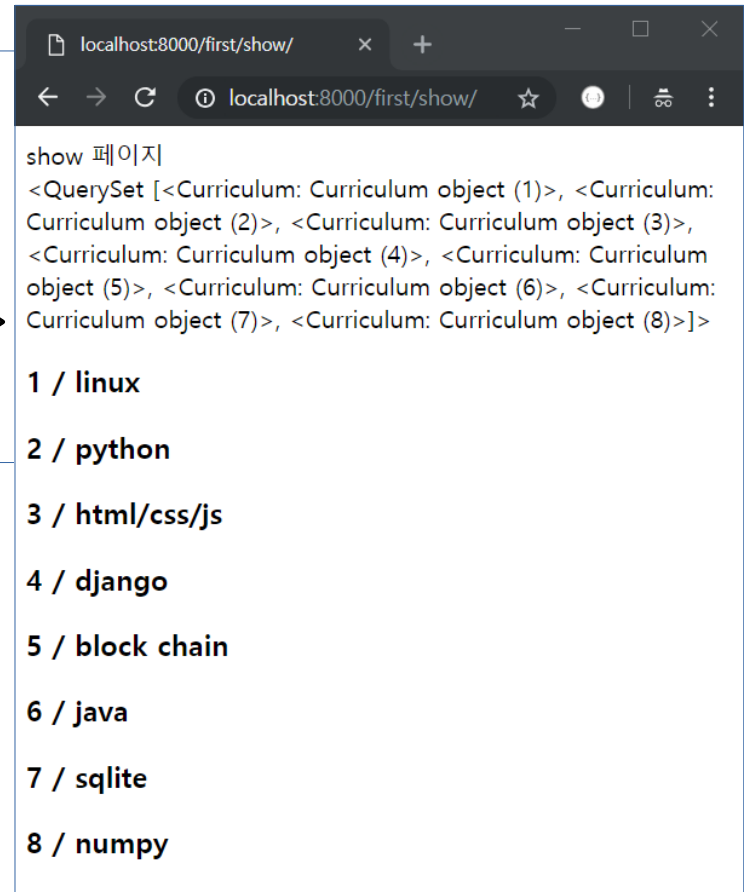


■ 템플릿 활용

● 템플릿 html 데이터 전달 - 각 요소 출력

- show.html

```
<body>
  show 페이지<br>
  {{ list }}<br>
  {% for item in list %}
    <h3>{{item.id}} / {{item.name}}</h3>
  {% endfor %}
</body>
```



■ MySQL DB 설정

● 필요 모듈 설치

– mysqlclient / pymysql

```
C:\Windows\system32\cmd.exe - conda install mysqlclient

C:\W>conda install mysqlclient
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\WGGoRebW\Miniconda3

  added / updated specs:
    - mysqlclient

The following packages will be downloaded:



| package                  | build          |         |
|--------------------------|----------------|---------|
| mysql-connector-c-6.1.11 | h33f27b4_1     | 1.8 MB  |
| mysqlclient-1.3.14       | py37he774522_0 | 1009 KB |
| Total:                   |                | 2.8 MB  |



The following NEW packages will be INSTALLED:

  mysql-connector-c  pkgs/main/win-64::mysql-connector-c-6.1.11-h33f27b4_1
  mysqlclient       pkgs/main/win-64::mysqlclient-1.3.14-py37he774522_0

Proceed ([y]/n)? _
```

■ MySQL DB 설정

● settings.py 접속 DB 환경설정

– DATABASES

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'django',  
        'USER': 'root',  
        'PASSWORD': '1234',  
        'HOST': 'localhost',  
        'PORT': 3306  
    }  
}
```