

■ 프로젝트 구조 생성

- django-admin startproject [프로젝트명] [생성경로]

명령 프롬프트

```
C:\dev\django>mkdir tutorial
```

```
C:\dev\django>cd tutorial
```

```
C:\dev\django\tutorial>django-admin startproject config .
```

```
C:\dev\django\tutorial>
```

- 생성 결과

```
W---tutorial
    |
    |   manage.py
    |
    W---config
        |
        |   asgi.py
        |   settings.py
        |   urls.py
        |   wsgi.py
        |   __init__.py
```

■ App 생성

● django-admin startapp [App명]

C:\> 명령 프롬프트

```
C:\dev\django\tutorial>django-admin startapp firstapp
```

```
python manage.py startapp firstapp
```

```
C:\dev\django\tutorial>
```

● 생성 결과

```
W---tutorial
|
|   manage.py
|
+---config
|   asgi.py
|   settings.py
|   urls.py
|   wsgi.py
|   __init__.py
|
W---firstapp
|   admin.py
|   apps.py
|   models.py
|   tests.py
|   views.py
|   __init__.py
|
W---migrations
|   __init__.py
```

■ 프로젝트 실행

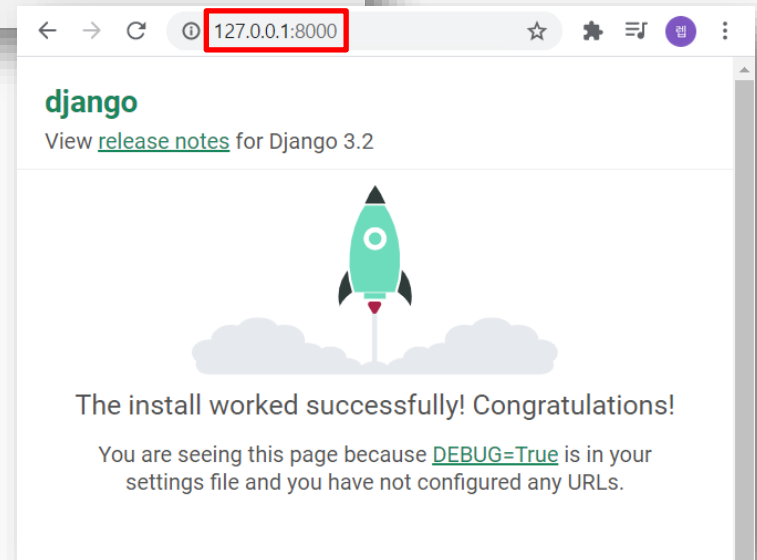
● python manage.py runserver [domain][:][port]

```
명령 프롬프트 - python manage.py runserver

C:\Wdev\Wdjango\tutorial>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

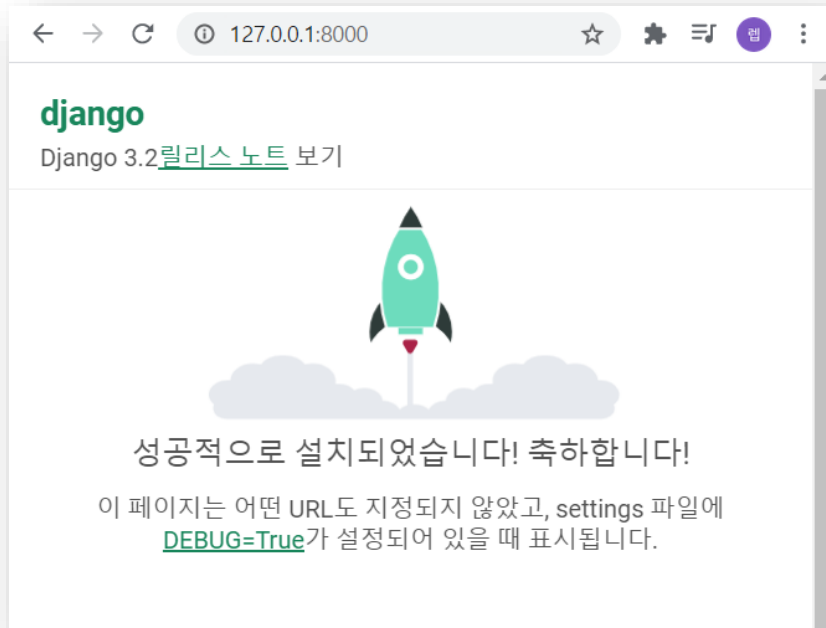
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 31, 2021 - 09:40:11
Django version 3.2.3, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



■ 언어 및 시간 설정

● settings.py

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



■ 웹페이지 주소 생성

- 사용자의 요청을 받은 후 처리하여 응답 - View

- **firstapp**/views.py

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

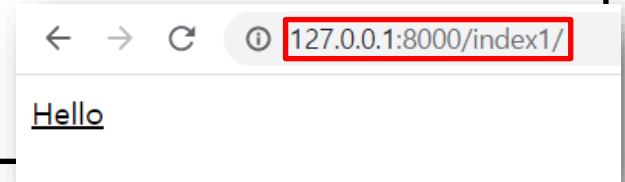
def index1(request):
    return HttpResponse('<u>Hello</u>')
```

- 사용자가 요청할 주소 생성 및 View 내의 함수와 연결 - Routing

- **config**/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),
]
```



■ 웹페이지 주소 생성

- 사용자의 요청을 받은 후 처리하여 응답 - View

- **firstapp/views.py**

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

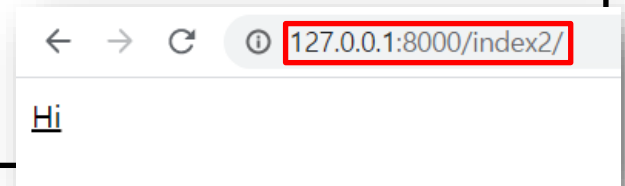
def index2(request):
    return HttpResponse('<u>Hi</u>')
```

- 사용자가 요청할 주소 생성 및 View 내의 함수와 연결 - Routing

- **config/urls.py**

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

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



■ 웹페이지 주소 생성

- 사용자의 요청을 받은 후 처리하여 응답 - View

- **firstapp**/views.py

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

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

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

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

■ 웹페이지 주소 생성

● config/urls.py → firstapp/urls.py

- **config/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')),
]
```

- **firstapp/urls.py**

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

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

← → ↻ ⓘ 127.0.0.1:8000/first/main/

Main

■ 데이터베이스 활용

- 데이터베이스에 생성(또는 조회) 될 테이블 내용 작성 - Model

- **firstapp/models.py**

```
from django.db import models

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

- **config/settings.py**

```
INSTALLED_APPS = [
    'firstapp',    프로젝트에 사용될 App명 등록, 목록의 상단에 입력 (가끔 static 오류 발생)
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

■ 데이터베이스 활용

● Model 적용

- python manage.py makemigrations [app명]

```
C:\dev\django\tutorial>python manage.py makemigrations firstapp
Migrations for 'firstapp':
  firstapp\migrations\0001_initial.py
    - Create model Curriculum

C:\dev\django\tutorial>
```

데이터베이스에 적용될 내용 정리

- python manage.py migrate

```
C:\dev\django\tutorial>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, firstapp, sessions

Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
```

데이터베이스에 적용

■ 데이터베이스 활용

● Model 적용

- 적용 결과

DB Browser for SQLite - C:\dev\django\tutorial\db.sqlite3

파일(F) 편집(E) 뷰(V) Tools 도움말(H)

새 데이터베이스(N) 데이터베이스 열기(O) 변경 사항 저장하기(W)

데이터베이스 구조 데이터 보기 Pragma 수정 SQL 실행

테이블 생성하기(C) 인덱스 생성하기(I) 테이블 수정하기 테이블 삭제하기 >>

이름	타입	스키마
▼ 테이블 (12)		
> auth_group		CREATE TABLE "auth_group" ("id" integer NOT NULL)
> auth_group_permissions		CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL)
> auth_permission		CREATE TABLE "auth_permission" ("id" integer NOT NULL)
> auth_user		CREATE TABLE "auth_user" ("id" integer NOT NULL)
> auth_user_groups		CREATE TABLE "auth_user_groups" ("id" integer NOT NULL)
> auth_user_user_permissions		CREATE TABLE "auth_user_user_permissions" ("id" integer NOT NULL)
> django_admin_log		CREATE TABLE "django_admin_log" ("id" integer NOT NULL)
> django_content_type		CREATE TABLE "django_content_type" ("id" integer NOT NULL)
> django_migrations		CREATE TABLE "django_migrations" ("id" integer NOT NULL)
> django_session		CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL)
▼ firstapp_curriculum		CREATE TABLE "firstapp_curriculum" ("id" integer NOT NULL)
id	integer	"id" integer NOT NULL
name	varchar(255)	"name" varchar(255) NOT NULL
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
> 인덱스 (15)		

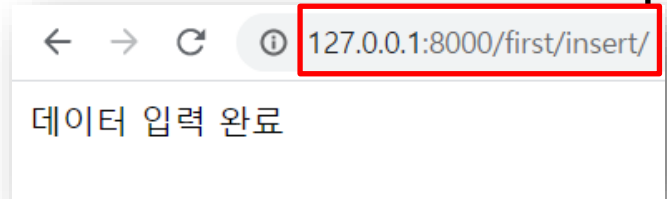
■ 데이터베이스 활용

● Model을 활용한 데이터 입력

- **firstapp**/views.py

```
from .models import Curriculum

def insert(request):
    # 1-linux 입력
    Curriculum.objects.create(name='linux')
    # 2-python 입력
    c = Curriculum(name='python')
    c.save()
    # 3-html/css/js 입력
    Curriculum(name='python').save()
    # 4-django 입력
    Curriculum(name='django').save()
    return HttpResponse('데이터 입력 완료')
```



- **firstapp**/urls.py

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

A screenshot of the Django admin interface. The top bar shows '데이터베이스 구조' (Database Structure) and '데이터 보기' (View Data). Below it, a dropdown menu shows '테이블(T): firstapp_curriculum'. The main area displays a table with 4 records. The table has columns 'id' and 'name'. The records are: 1 linux, 2 python, 3 python, and 4 django. The table is highlighted with a red box.

	id	name
...	필터	
1	1	linux
2	2	python
3	3	python
4	4	django

■ 데이터베이스 활용

● Model을 활용한 데이터 조회

- **firstapp**/views.py

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

- **firstapp**/urls.py

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

← → ↻ ⓘ 127.0.0.1:8000/first/show/

linux
python
python
django

■ 관리자 기능 활용

● 관리자 정보 등록

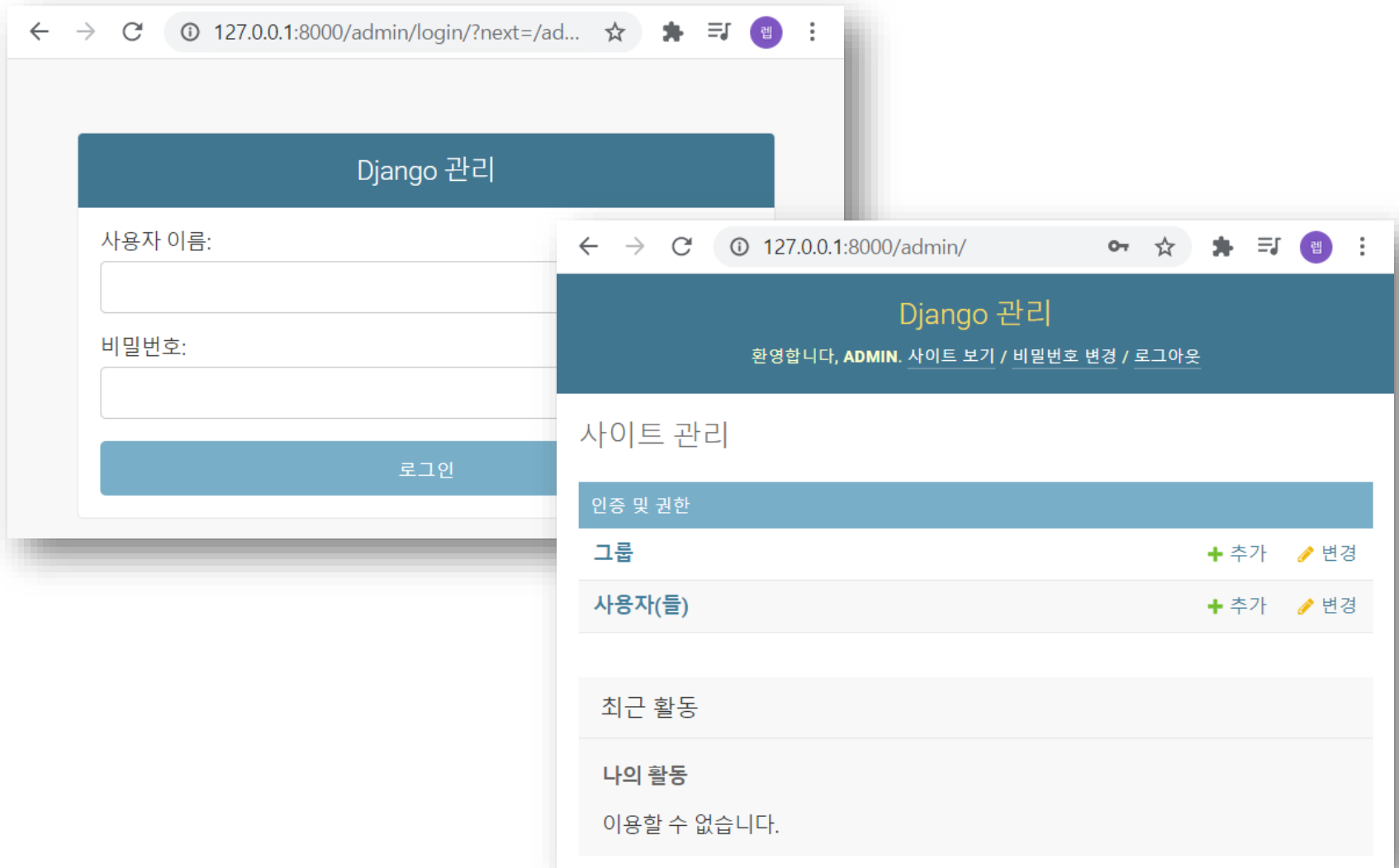
- python manage.py createsuperuser

```
C:\dev\django\tutorial>python manage.py createsuperuser
사용자 이름 (leave blank to use 'ggoreb'): admin
이메일 주소: admin@ggoreb.com
Password:
Password (again): 비밀번호는 입력된 문자가 보이지 않음
비밀번호가 너무 짧습니다. 최소 8 문자를 포함해야 합니다.
비밀번호가 너무 일상적인 단어입니다.
비밀번호가 전부 숫자로 되어 있습니다.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

■ 관리자 기능 활용

● 관리자 사이트

- `http://127.0.0.1:8000/admin`



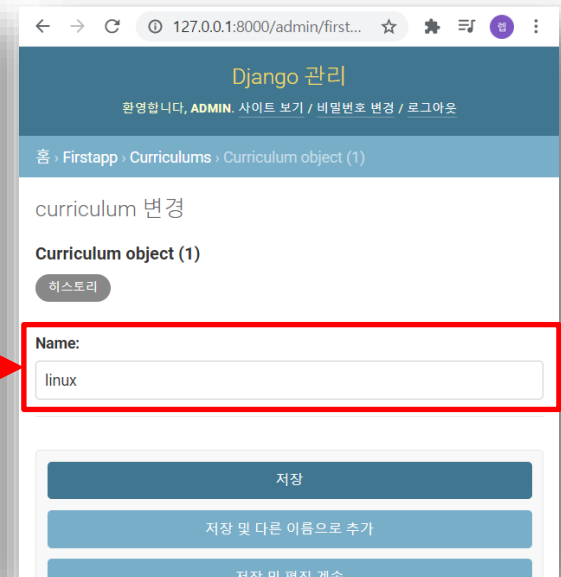
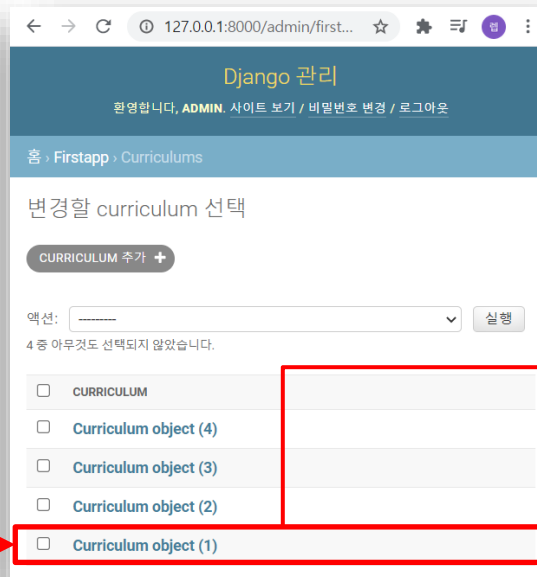
■ 관리자 기능 활용

- 관리자 사이트에서 데이터를 제어 할 수 있도록 등록

- **firstapp/admin.py**

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

admin.site.register(Curriculum)
```



■ Template 활용

- 사용자에게 보여주는 응답 결과 파일 사용 - Template

- **firstapp/templates/firstapp/show.html**

```
<body>
  <h1>show!</h1>
</body>
```

- **firstapp/views.py**

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

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

- > config
- ▼ firstapp
 - > __pycache__
 - > migrations
 - ▼ templates \ firstapp
 - <> show.html
- 🔗 __init__.py
- 🔗 admin.py
- 🔗 apps.py
- 🔗 models.py
- 🔗 tests.py
- 🔗 urls.py
- 🔗 views.py
- ☰ db.sqlite3
- 🔗 manage.py

← → ↻ ⓘ 127.0.0.1:8000/first/show/ ☆ ⚙ ⋮

show!

■ Template 활용

- 사용자에게 보여주는 응답 결과 파일 사용 - Template

- **firstapp/templates/firstapp/show.html**

```
<body>
  <h1>show!</h1>
  <hr>
  {{ data }}
</body>
```

- **firstapp/views.py**

```
def show(request):
    curriculum = Curriculum.objects.all()

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

← → ↻ ⓘ 127.0.0.1:8000/first/show/ ☆ ⚙ ⋮ 앱 ⋮

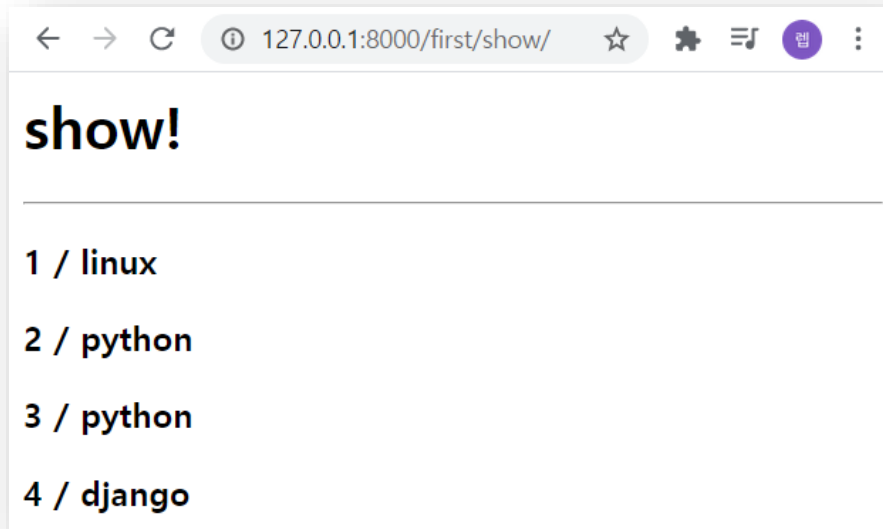
show!

<QuerySet [<Curriculum: Curriculum object (1)>, <Curriculum: Curriculum object (2)>, <Curriculum: Curriculum object (3)>, <Curriculum: Curriculum object (4)>]>

■ Template 활용

- 사용자에게 보여주는 응답 결과 파일 사용 - Template
 - **firstapp/templates/firstapp/show.html**

```
<body>
  <h1>show!</h1>
  <hr>
  {% for item in data %}
    <h3>{{ item.id }} / {{ item.name }}</h3>
  {% endfor %}
</body>
```



■ 데이터베이스 설정 (MySQL 사용 시)

- 필요한 모듈 설치 : `pip install mysqlclient`
- 데이터베이스 정보 입력 및 SQL 로그 출력

- **config/settings.py**

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

LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'handlers': {
        'console': {
            'level': 'DEBUG',
            'class': 'logging.StreamHandler',
        }
    },
    'loggers': {
        'django.db.backends': {
            'handlers': ['console'],
            'level': 'DEBUG',
        },
    },
}
```