

Django CRUD 실습

# 8번째 세션

---

NEXT X LIKELION 강미지

# 드디어 CRUD를 다 배웠습니다~!

create & read & update & delete



CREATE



READ



UPDATE



DELETE

---

C

R

U

D

# 오늘의 실습

전공 과목 블로그

**http://127.0.0.1:8000/**

등록한 모든 전공과 모든 과목을 보여줍니다

## 모든 전공

### 전공 추가

- 컴퓨터학과
- 중어중문학과
- 사회학과
- 영어영문학과
- 생명과학부
- 멋사학과

## 모든 과목

### 과목 추가

1. 자료구조 | 컴퓨터학과 | 이도길  
자료구조 수업
2. 알고리즘 | 컴퓨터학과 | 이도길  
과제 해야 돼요
3. Git&Github | 멋사학과 | 전해린  
덕분에 Git 마스터했어요
4. Django | 멋사학과 | 김범진  
듣고 CRUD랑 친해졌어요

# 오늘의 실습

전공 과목 블로그

**http://127.0.0.1:8000/addMajor**

전공을 추가할 수 있습니다.

← → ↻ ⓘ 127.0.0.1:8000/addMajor

## 전공 추가

Name:

등록

**http://127.0.0.1:8000/addSubject**

과목을 추가할 수 있습니다.

← → ↻ ⓘ 127.0.0.1:8000/addSubject

## 과목 추가

Major:  ▼

Subject name:

Prof name:

Memo:

등록

# 오늘의 실습

전공 과목 블로그

**http://127.0.0.1:8000/computer**

해당 전공 과목만 filtering 할 수 있습니다.

← → ↻ ⓘ 127.0.0.1:8000/computer

## 컴과 과목

1. 자료구조 | 컴퓨터학과 | 이도길 | [수정](#) [삭제](#)

자료구조 수업

2. 알고리즘 | 컴퓨터학과 | 이도길 | [수정](#) [삭제](#)

과제 해야 돼요

**http://127.0.0.1:8000/likelion**

해당 전공 과목만 filtering 할 수 있습니다.

← → ↻ ⓘ 127.0.0.1:8000/likelion

## 멋사 과목

1. Git&Github | 멋사학과 | 전해린 | [수정](#) [삭제](#)

덕분에 Git 마스터했어요

2. Django | 멋사학과 | 김범진 | [수정](#) [삭제](#)

듣고 CRUD랑 친해졌어요

# 오늘의 실습

전공 과목 블로그

## http://127.0.0.1:8000/editSubject

특정 과목 정보를 수정할 수 있습니다.

← → ↻ ⓘ 127.0.0.1:8000/editSubject/8

## 정보 수정하기

Major:

Subject name:

Prof name:

Memo:

## http://127.0.0.1:8000/deleteSubject

특정 과목을 삭제할 수 있습니다.

← → ↻ ⓘ 127.0.0.1:8000/likelion

## 멋사 과목

1. Git&Github | 멋사학과 | 전해린 | [수정](#) [삭제](#)

덕분에 Git 마스터했어요

2. Django | 멋사학과 | 김범진 | [수정](#) [삭제](#)

듣고 CRUD랑 친해졌어요

# 앱을 생성해 봅시다

전공 과목 블로그

```
$ mkdir session8
```

```
$ cd session8
```

```
$ pipenv shell
```

```
$ pipenv install django
```

```
$ django-admin startproject subjectProject
```

```
$ cd subjectProject
```

```
$ python manage.py startapp subjectApp
```

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'subjectApp',  
]
```

Vscode의 settings.py의 INSTALLED\_APPS에 subjectApp추가

# DB와 관리자 계정을 생성해 봅시다

전공 과목 블로그

```
$ python manage.py makemigrations
```

```
$ python manage.py migrate
```

```
$ python manage.py createsuperuser
```

```
$ winpty python manage.py createsuperuser
```



# 실습 1단계 – 전공 CR

전공 과목 블로그

← → ↻ ⓘ 127.0.0.1:8000/addMajor

## 전공 추가

Name:

등록

## 모든 전공

### 전공 추가

- 컴퓨터학과
- 중어중문학과
- 사회학과
- 영어영문학과
- 생명과학부
- 멋사학과

# 실습 1단계 – 전공 CR

전공 과목 블로그

← → ↻ ⓘ 127.0.0.1:8000/addMajor

## 전공 추가

Name:

등록

우선 Major 모델을 만들어 봅시다!

field에는 이름만 있으면 될 것 같아요

Model 변경 후 makemigrations & migrate 잊지 말고 해주세요

# 실습 1단계 – 전공 CR

전공 과목 블로그

← → ↺ ⓘ 127.0.0.1:8000/addMajor

## 전공 추가

Name:

등록

다음으로 전공 추가 페이지를 만듭시다

# 실습 1단계 – 전공 CR

전공 과목 블로그

이런식으로 만들었었죠

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Document</title>
9 </head>
10
11 <body>
12   <h2>저공 추가</h2>
13   <form action="/addMajor" method="post">
14     {% csrf_token %}
15     <div>
16       <div>전공 이름</div>
17       <input type="text" name="name" />
18     </div>
19     <button type="submit">등록</button>
20   </form>
21
22 </body>
23
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Document</title>
8   </head>
9   <body>
10     <h2>글 쓰기 페이지</h2>
11     <form action="/new" method="post">
12       {% csrf_token %}
13       <div>
14         <div>글 제목</div>
15         <input type="text" name="title" />
16       </div>
17       <div>
18         <div>글 내용</div>
19         <textarea name="content" cols="30" rows="10"></textarea>
20       </div>
21       <button type="submit">작성 완료</button>
22     </form>
23   </body>
24 </html>
```

# 실습 1단계 – 전공 CR

전공 과목 블로그

추가할 필드가 많아질 수록 코드가 많이 늘어나네요

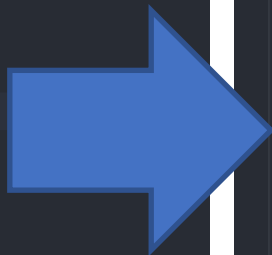
```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Document</title>
9 </head>
10
11 <body>
12   <h2>전공 추가</h2>
13   <form action="/addMajor" method="post">
14     {% csrf_token %}
15     <div>
16       <div>전공 이름</div>
17       <input type="text" name="name" />
18     </div>
19     <button type="submit">등록</button>
20   </form>
21
22 </body>
23
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Document</title>
8   </head>
9   <body>
10     <h2>글 쓰기 페이지</h2>
11     <form action="/new" method="post">
12       {% csrf_token %}
13       <div>
14         <div>글 제목</div>
15         <input type="text" name="title" />
16       </div>
17       <div>
18         <div>글 내용</div>
19         <textarea name="content" cols="30" rows="10"></textarea>
20       </div>
21       <button type="submit">작성 완료</button>
22     </form>
23   </body>
24 </html>
```

# 어느정도 자동으로 form을 만들 수는 없을까

forms.py

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Document</title>
9 </head>
10
11 <body>
12     <h2>전공 추가</h2>
13     <form action="/addMajor" method="post">
14         {% csrf_token %}
15         <div>
16             <div>전공 이름</div>
17             <input type="text" name="name" />
18         </div>
19         <button type="submit">등록</button>
20     </form>
21
22 </body>
23
```



```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

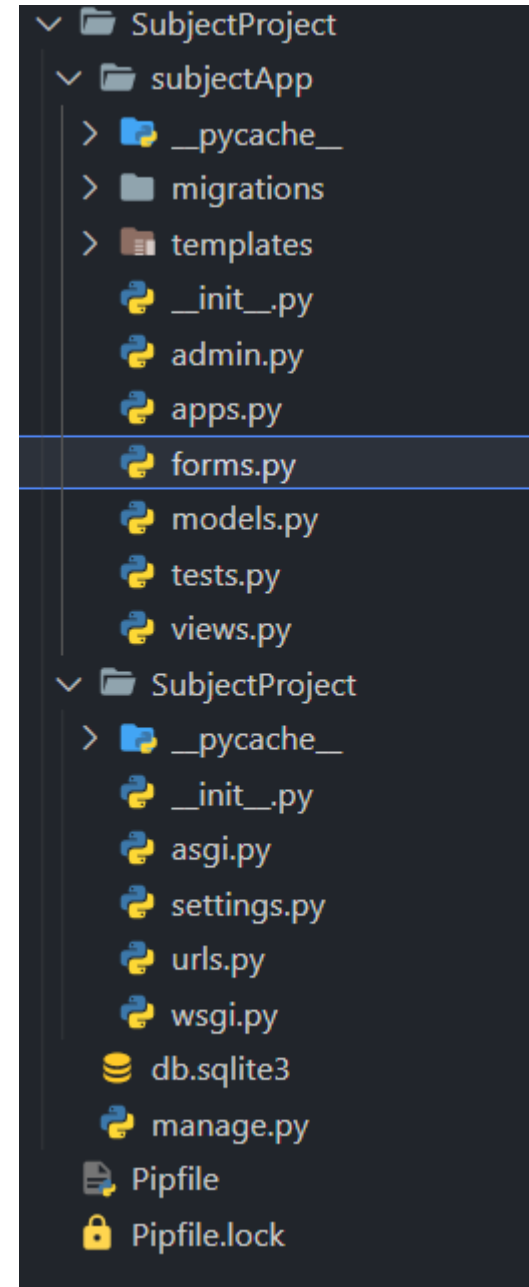
<body>
    <h2>전공 추가</h2>
    <div>
        <form method="POST">
            {% csrf_token %}
            {{ form.as_p }}
            <button>등록</button>
        </form>
    </div>
</body>

</html>
```

# forms.py를 이용해서 추가 페이지 만들기

create

0. Major model 생성(M) – 이미 했습니다
1. subjectApp 하단에 forms.py 생성
2. addMajor.html 생성 (T)
3. views.py에 forms.py를 이용해서 view작성
4. urls.py에 해당 url 생성



# 1. subjectApp 하단에 forms.py 생성

Form을 작성해 봅시다

```
from django import forms
from .models import Major
```

```
class MajorModelForm(forms.ModelForm):
    class Meta:
        model = Major
        fields = ('name',)
        widgets = {
            'name': forms.TextInput(attrs={'class': 'form-control', 'placeholder': '전공을 입력하세요.'}),
        }
```



## 2. addMajor.html 생성 (T)

코드가 짧아졌네요

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>
  <h2>전공 추가</h2>
  <div>
    <form method="POST">
      {% csrf_token %}
      {{ form.as_p }}
      <button>등록</button>
    </form>
  </div>
</body>

</html>
```

### 3. views.py에 forms.py를 이용해서 view작성

Form을 이용하기로 했으니 CreateView를 이용합시다

```
from django.shortcuts import render, redirect
from .models import Subject, Major
from django.views.generic import CreateView, UpdateView
from .forms import MajorModelForm
from django.urls import reverse_lazy
```

```
class AddMajorView(CreateView):
    model = Major
    form_class = MajorModelForm
    template_name = 'addMajor.html'
    success_url = reverse_lazy('home')
```

## 4. urls.py에 해당 url 생성

urls.py 약간 방법이 달라졌어요

```
from django.contrib import admin
from django.urls import path
from subjectApp import views
from subjectApp.views import AddMajorView
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('addMajor', AddMajorView.as_view(), name="addMajor"),
]
```

# 진짜 실습 1단계 – 전공 CR (시간상 UD는 생략..일까요)

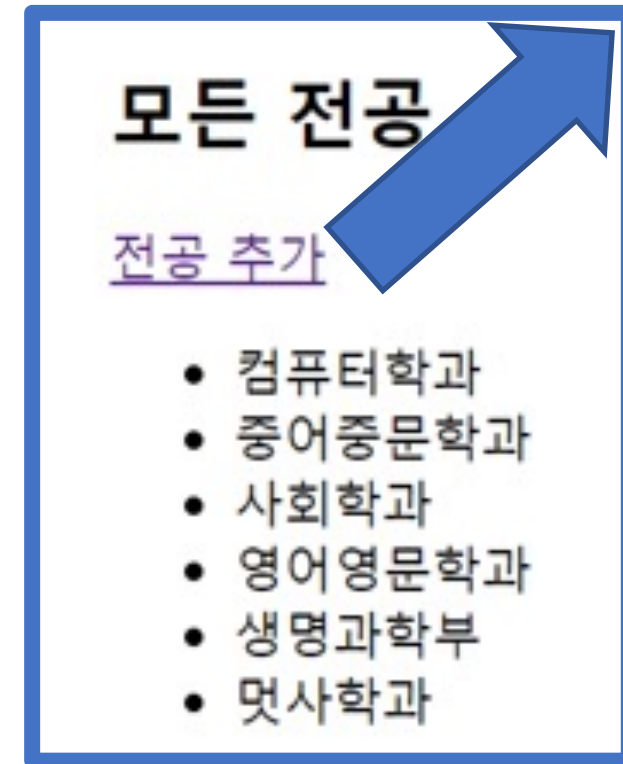
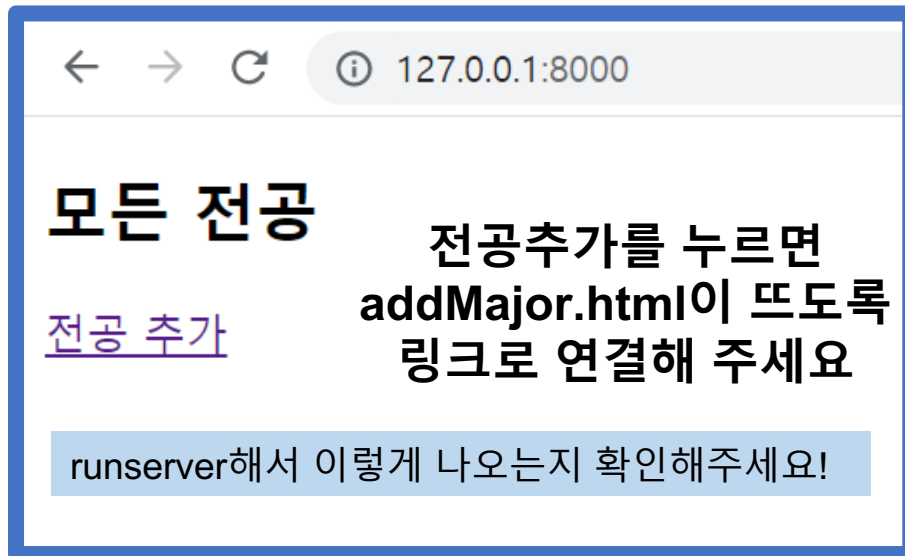
read

추가한 전공들을 보기 위한 페이지를 만듭시다.

할일

1. home.html 생성
2. Views.py 에 home함수 작성
3. Url 연결

+ home.html에서  
전공 추가를 클릭하면  
addMajor.html로 연결되게 링크를 걸어주세요



전공을 추가하면 이렇게 됩니다.

# 실습 2단계 – 과목 CRUD

home.html

read

추가한 과목들을 보기 위한 페이지를 만듭시다.  
우선 Read 부터!

## 할일

1. Subject model 생성
2. home.html에 코드 추가하기
3. Views.py의 home 함수에 코드 추가하기
4. Url 연결 – 이미 했습니다

Subject 라는 모델에는  
과목이름/소속학과/교수님이름/메모가 필요할 것 같아요

소속학과는 Major 모델을 foreign key로 사용해 봅시다

## 모든 전공

### 전공 추가

- 컴퓨터학과
- 중어중문학과
- 사회학과
- 영어영문학과
- 생명과학부
- 멋사학과

## 모든 과목

### 과목 추가

1. 자료구조 | 컴퓨터학과 | 이도길  
자료구조 수업
2. 알고리즘 | 컴퓨터학과 | 이도길  
과제 해야 돼요
3. Git&Github | 멋사학과 | 전해린  
덕분에 Git 마스터했어요
4. Django | 멋사학과 | 김범진  
듣고 CRUD랑 친해졌어요

## 실습 2단계 – 과목 CRUD

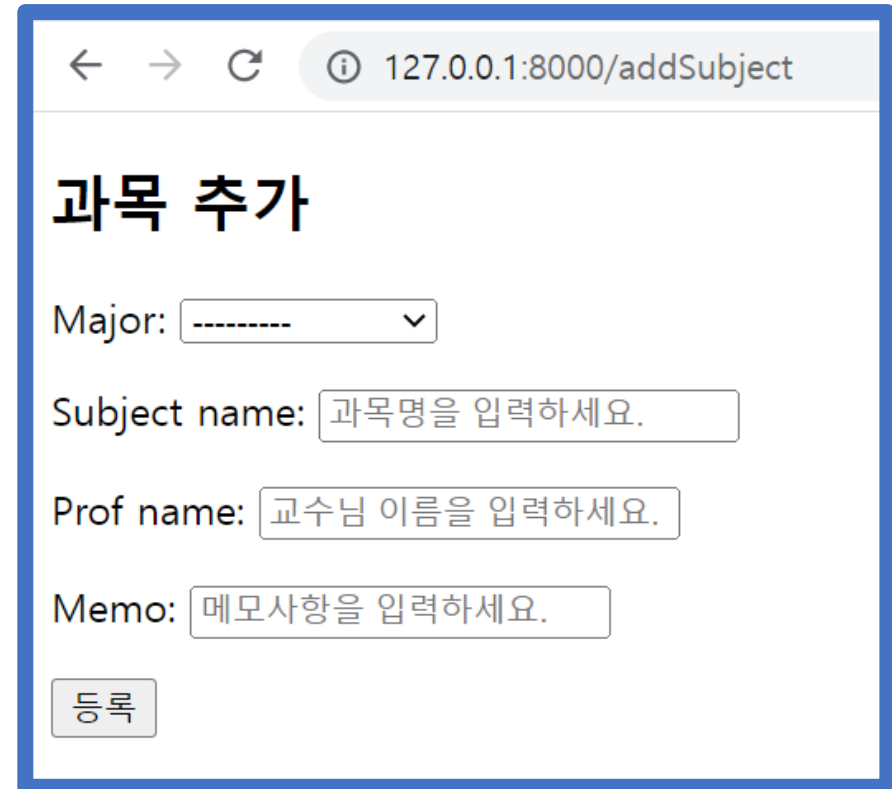
create

이제 과목을 추가할 수 있는 페이지를 만듭시다.

### 할일

1. Subject model 생성 – 이미 했습니다
2. forms.py에 SubjectModelForm 작성하기  
- 같이 해요
3. addSubject.html 작성하기  
아까 만든 addMajor.html을 참고해서 만들어 보세요
3. Views.py에 AddSubjectView 만들기  
아까 만든 AddMajorView를 참고해서 만들어 보세요
4. Url 연결

+ 아까 만든 home.html에서  
과목 추가를 클릭하면  
addSubject.html로 연결되게 링크를 걸어주세요



← → ↻ ⓘ 127.0.0.1:8000/addSubject

### 과목 추가

Major:

Subject name:

Prof name:

Memo:

## 실습 2단계 – 과목 CRUD

forms.py 처음이니까 따라 쳐 봅시다~

```
from django import forms
from .models import Subject, Major

Major_choices = Major.objects.all().values_list('name', 'name')

class MajorModelForm(forms.ModelForm):
    class Meta:
        model = Major
        fields = ('name',)
        widgets = {
            'name': forms.TextInput(attrs={'class': 'form-control', 'placeholder': '전공을 입력하세요.'}),
        }

class SubjectModelForm(forms.ModelForm):
    class Meta:
        model = Subject
        fields = ('major', 'subject_name', 'prof_name', 'memo')
        widgets = {
            'major': forms.Select(choices=Major_choices, attrs={'class': 'form-control'}),
            'subject_name': forms.TextInput(attrs={'class': 'form-control', 'placeholder': '과목명을 입력하세요.'}),
            'prof_name': forms.TextInput(attrs={'class': 'form-control', 'placeholder': '교수님 이름을 입력하세요.'}),
            'memo': forms.TextInput(attrs={'class': 'form-control', 'placeholder': '메모사항을 입력하세요.'}),
        }
```

## 실습 2단계 – 과목 CRUD

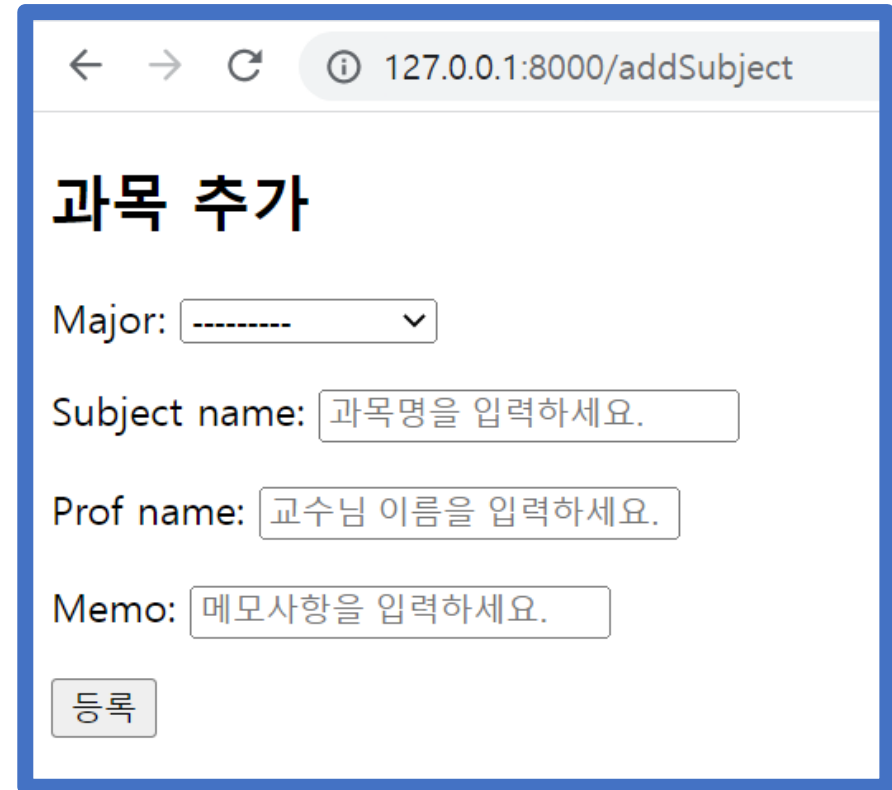
create

이제 과목을 추가할 수 있는 페이지를 만듭시다.

### 할일

1. Subject model 생성 – 이미 했습니다
2. forms.py에 SubjectModelForm 작성하기  
-완료
3. addSubject.html 작성하기  
아까 만든 addMajor.html을 참고해서 만들어 보세요
3. Views.py에 AddSubjectView 만들기  
아까 만든 AddMajorView를 참고해서 만들어 보세요
4. Url 연결

+ 아까 만든 home.html에서  
과목 추가를 클릭하면  
addSubject.html로 연결되게 링크를 걸어주세요



← → ↻ ⓘ 127.0.0.1:8000/addSubject

### 과목 추가

Major:

Subject name:

Prof name:

Memo:



## 실습 2단계 – 과목 CRUD

create

← → ↻ ⓘ 127.0.0.1:8000/addSubject

### 과목 추가

Major:  ▼

Subject name:

Prof name:

Memo:

← → ↻ ⓘ 127.0.0.1:8000

### 모든 전공

[전공 추가](#)

### 모든 과목

[과목 추가](#)

runserver해서 이렇게 나오는지 확인해주세요!

## 실습 2단계 – 쉬는 시간

create

전공 2개 이상

과목은 3개정도 추가해봅시다~

단, 과목은 전부 같은 전공 과목이면 안됩니다.

← → ↻ ⓘ 127.0.0.1:8000/addSubject

### 과목 추가

Major:  ▼

Subject name:

Prof name:

Memo:

## 모든 전공

### 전공 추가

- 컴퓨터학과
- 중어중문학과
- 사회학과
- 영어영문학과
- 생명과학부
- 멋사학과

## 모든 과목

### 과목 추가

1. 자료구조 | 컴퓨터학과 | 이도길  
자료구조 수업
2. 알고리즘 | 컴퓨터학과 | 이도길  
과제 해야 돼요
3. Git&Github | 멋사학과 | 전해린  
덕분에 Git 마스터했어요
4. Django | 멋사학과 | 김범진  
듣고 CRUD랑 친해졌어요

## 실습 2단계 – 과목 CRUD

read

나는 컴퓨터학과 수업 전공만 보고 싶은데...

할일

1. computer.html 작성하기
3. Views.py에 computerSubjectView 함수 만들기
4. Url 연결

← → ↻ ⓘ 127.0.0.1:8000/computer

### 컴과 과목

1. 자료구조 | 컴퓨터학과 | 이도길 | [수정](#) [삭제](#)

자료구조 수업

2. 알고리즘 | 컴퓨터학과 | 이도길 | [수정](#) [삭제](#)

과제 해야 돼요

## 실습 2단계 – 과목 CRUD

update

이제 과목을 수정할 수 있는 페이지를 만듭시다.

할일

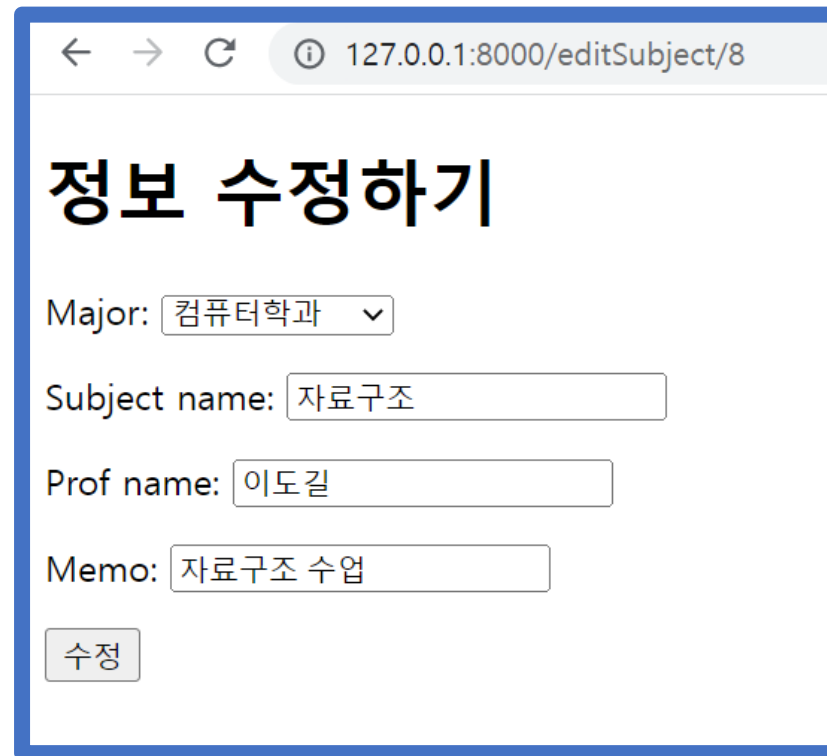
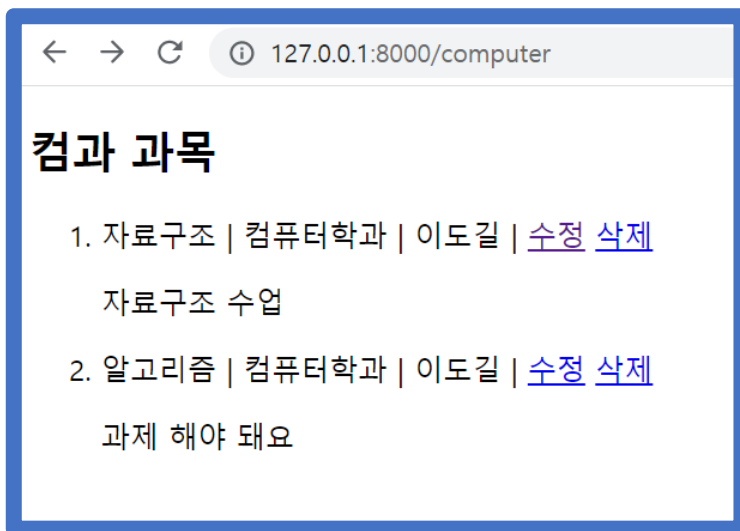
### 1. editSubject.html 작성하기

좀전에 만든addSubject.html과 거의 동일합니다.

### 3. Views.py에 EditSubjectView 만들기

이전에 만든AddSubjectView와 거의 유사해요  
CreateView대신 UpdateView 사용하면 됩니다~

### 4. Url 연결



+ 아까 만든 computer.html에서  
수정을 클릭하면  
editSubject.html로 연결되게 링크를 걸어주세요  
home.html로 redirect시켜주세요

## 실습 2단계 – 과목 CRUD

delete

이제 과목을 삭제할 수 있도록 해봅시다.

### 할일

#### 1. Views.py에 DeleteSubjectView 함수 만들기

이전처럼 만들고 싶다면 DeleteView를 사용하면 되고  
기존처럼 함수형식으로 만들어도 됩니다.

#### 2. Url 연결

+ 아까 만든 computer.html에서  
삭제를 클릭하면  
과목 정보가 삭제되고  
home.html로 redirect되게 해주세요

← → ↻ ⓘ 127.0.0.1:8000/computer

### 컴과 과목

1. 자료구조 | 컴퓨터학과 | 이도길 | [수정](#) [삭제](#)  
자료구조 수업
2. 알고리즘 | 컴퓨터학과 | 이도길 | [수정](#) [삭제](#)  
과제 해야 돼요

runserver해서 이렇게 나오는지 확인해주세요!

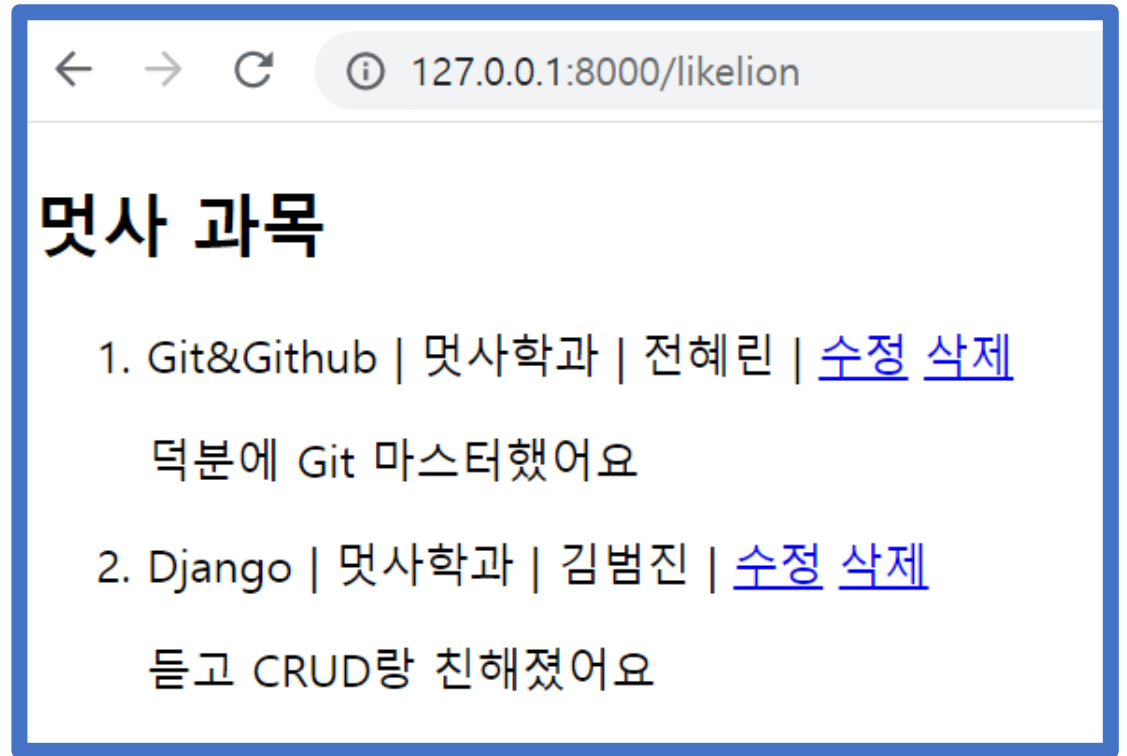
# 마지막! 실습 3단계 – 과목 CRUD

다른 전공 과목 보는 페이지는 왜 없지...

컴퓨터학과 외에 추가했던 전공 하나를 골라서  
그 전공의 과목만 보여주는 페이지를 만들어주세요.

할 일

Template 생성/ View작성 / URL연결



runserver해서 이렇게 나오는지 확인해주세요!



# 과제

아까 예고했듯이..

진짜 실습 1단계 - 전공 CR (시간상 UD는 생략..일까요)

read

**전공도 수정 삭제 할 수 있도록 페이지를 고쳐주세요 (필수)**

->삭제 시 다른 페이지를 거쳐서 삭제되게 해야 리다이렉션 문제가 생기지 않습니다.

**CSS로 페이지를 예쁘게 꾸며주세요 (필수)**

처음부터 앱을 다시 만들면 아주 좋겠지만 시험기간이니 이건 선택으로 남겨두겠습니다.

그래도 다음 세션 전까지 MTV패턴 복습은 꼭 하고 와주세요

그래도 나는 시간이 남는다 하시는 분들을 위해



# 과제

나는 시험 안보는데

**그래도 나는 시간이 남는다 하시는 분들을 위한 추가 과제 ♡ (선택입니다)**

저희는 전공 2개를 추가했고 전공별 과목을 보기 위한 페이지 2개를 만들었습니다.

우리 학교 전공이 대충 60개 정도 되는데요 (이중 융합 합치면 더 많겠죠?)  
60개 전공을 위한 세부페이지를 하나하나 만들면 너무 괴롭고 코드 낭비겠죠?  
(html 60개... view 60개... url 60개...)

그래서 과제는!

지금까지 만든 앱을 `querystring`을 이용해서 중복되는 코드를 줄이고 `template` 개수를 줄여보세요