

멋쟁이 사자처럼 2주차 강의 인스타그램 만들기

강의자:노기진

이번시간에 할 내용

- media 파일 다루기
- list 페이지
- 정렬(최신순, 좋아요)
- 프로필 사진 동그랗게 만들기

저번시간에 한 내용 바탕으로
다시 runserver를 해봅시다.

```
source myvenv/scripts/activate  
python manage.py runserver
```

혹시 이런 오류 뜨는 사람은,

```
django.db.migrations.exceptions.NodeNotFoundError: Migration insta.0001_initial dependencies reference nonexistent parent node ('auth', '0011_update_proxy_permissions')
```

```
python -m venv myvenv  
source myvenv/scripts/activate  
pip install Django  
pip install Pillow  
python manage.py runserver
```

1주 강의 끝

지금 이 상태일거예요!

main.html 를 수정해봅시다.

@ | CBNU

CBNU Instagram

Welcome to CBNU Instagram

```
views.py  base.html  main.html x
instagram > insta > templates > insta > main.html > ...
1  {% extends "insta/base.html" %}
2  {% block content %}
3
4  <!-- 점보트론 시작 -->
5  <div class="jumbotron">
6      <h1 class="display-4">CBNU Instagram</h1>
7      <p class="lead">Welcome to CBNU Instagram</p>
8  </div>
9  <!-- 점보트론 끝 -->
10
11 {% endblock %}
```

<https://getbootstrap.com/docs/4.3/components/jumbotron/>

bootstrap→jumbotron 첫번째 요소 사용



```
<!-- 정렬 박스-->
<div class="btn-group float-right">
  <div class="d-flex">
    <div class="btn-group">
      <button type="button" class="btn btn-secondary">정렬</button>
      <button type="button" class="btn btn-secondary dropdown-toggle dropdown-toggle-split" id="dropdownMenuReference" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" data-reference="parent">
        <span class="sr-only">Toggle Dropdown</span>
      </button>
      <div class="dropdown-menu" aria-labelledby="dropdownMenuReference">
        <a class="dropdown-item" href="?sort=new">최신</a>
        <a class="dropdown-item" href="?sort=like">좋아요</a>
      </div>
    </div>
  </div>
</div>
<!-- 정렬 박스 끝-->
```

<https://getbootstrap.com/docs/4.3/components/dropdowns/#dropdown-options>

bootstrap→dropdown options


```
<a class="dropdown-item" href="?:sort=new">최신</a>  
<a class="dropdown-item" href="?:sort=like">좋아요</a>
```

쿼리스트링 : 사용자가 웹프로그램으로 입력 데이터를 전달하는 방법

http://hostname/folder/file? *변수1=값1&변수2=값2&변수3=값3*

: ? 뒤의 부분

sort(변수)=new(값)

view로 보냄!

지금까지 잘 했으면 보이는 화면!

@CBNU

CBNU Instagram

Welcome to CBNU Instagram

정렬 ▼

내가 쓴 글을 보고싶다면?

`views.py` 를 수정해봅시다.

1. Dictionary (dict)

Dictionary는 "키(Key) - 값(Value)" 쌍을 요소로 갖는 컬렉션이다. Dictionary는 흔히 Map 이라고도 불리우는데, 키(Key)로 신속하게 값(Value)을 찾아내는 해시테이블(Hash Table) 구조를 갖는다.

파이썬에서 Dictionary는 "dict" 클래스로 구현되어 있다. Dictionary의 키(key)는 그 값을 변경할 수 없는 Immutable 타입이어야 하며, Dictionary 값(value)은 Immutable과 Mutable 모두 가능

```
import operator
```

```
dict={'A':1,'D':4,'C':3,'B':2}
```

```
sdct=sorted(dict.items(),key=operator.itemgetter(0))
```

#items() :key,value 쌍 얻는 함수

#인자값에 있는 key=operator.itemgetter(0)는
정렬하고자 하는 키 값을 0번째 인덱스 기준으로 하겠다.

#0번째 인자는 key임.

```
import operator
```

```
dict={'A':1,'D':4,'C':3,'B':2}
```

```
sdct=sorted(dict.items(),key=operator.itemgetter(1))
```

#items() :key,value 쌍 얻는 함수

#인자값에 있는 key=operator.itemgetter(1)는
정렬하고자 하는 값을 1번째 인덱스 기준으로 하겠다.

#1번째 인자는 value임.

```
import operator
```

```
dict={'A':1,'D':4,'C':3,'B':2}
```

```
sdict=sorted(dict.items(),key=operator.itemgetter(0))
```

```
dict={'A':1,'D':4,'C':3,'B':2}
```

```
sdict=sorted(dict.items(),key=operator.itemgetter(1))
```

실행결과는?

insta >  views.py >  main

```
1  from django.shortcuts import render
2  from .models import Post
3  import operator
4
5  # Create your views here.
6  def main(request):
7      posts = Post.objects.all()
8      sort = request.GET.get('sort', '')
9
10     if sort == 'new':
11         posts = Post.objects.all()
12     elif sort == 'like':
13         ordered_posts = {}
14         post_list = Post.objects.all()
15         for post in post_list:
16             ordered_posts[post] = post.like_count
17         post_list = sorted(ordered_posts.items(), key=operator.itemgetter(1), reverse=True)
18         posts = []
19         for post in post_list:
20             posts.append(post[0])
21
22     return render(request, 'insta/main.html', {
23         'posts': posts,
24     })
```


insta > views.py > main

```
1 from django.shortcuts import render
2 from .models import Post
3 import operator
4
5 # Create your views here.
6 def main(request):
7     posts = Post.objects.all()
8     sort = request.GET.get('sort', '')
9
10    if sort == 'new':
11        posts = Post.objects.all()
12    elif sort == 'like':
13        ordered_posts = {}
14        post_list = Post.objects.all()
15        for post in post_list:
16            ordered_posts[post] = post.like_count
```

내림차순으로 정렬

```
href="?sort=new">최신</a>
href="?sort=like">좋아요</a>
```

```
class Meta:
    ordering = ['-updated']
```

```
elif sort == 'like':
```

```
    ordered_posts = {}
```

```
    post_list = Post.objects.all()
```

```
    for post in post_list:
```

```
        | ordered_posts[post] = post.like_count
```

```
    post_list = sorted(ordered_posts.items(), key=operator.itemgetter(1), reverse=True)
```

```
    posts = []
```

```
    for post in post_list:
```

```
        | posts.append(post[0])
```

```
    return render(request, 'insta/main.html', {
```

```
        'posts': posts,
```

```
    })
```

```
@property
```

```
def like_count(self):
```

```
    return self.like_set.count()
```

{post.id : 좋아요갯수 } 인 dict형

ordered_posts 에 dict 형식을 쓴 이유?
post.id랑 post.좋아요를 짝지으려고!

reverse=True는 내림차순 정렬

```
(<Post: Post object (1)>, 4)
```

```
12 elif sort == 'like':
13     ordered_posts = {}
14     post_list = Post.objects.all()
15     for post in post_list:
16         ordered_posts[post] = post.
17     post_list = sorted(ordered_post
18     posts = []
19     for post in post_list:
20         posts.append(post[0])
```

(<Post: Post object (1)>, 4)

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

쓴 글을 보고싶다면 ?

이번 글에는 사진파일까지 같이 있는데 ?

Media 파일

웹서비스 이용자들이 업로드 하는 파일

```
class Post(models.Model):  
    user=models.ForeignKey(User,on_delete=models.CASCADE) #  
    image=models.ImageField()
```

settings.py 에서
MEDIA_URL
MEDIA_ROOT
추가

```
125 MEDIA_URL='/media/'  
126 MEDIA_ROOT=os.path.join(BASE_DIR, 'media')
```

```
125 MEDIA_URL='/media/'  
126 MEDIA_ROOT=os.path.join(BASE_DIR, 'media')
```

`MEDIA_URL='/media/'`

각 media 파일에 대한 URL의 고정값을 설정할 수 있음.

ex)필드명.url 에 의해 MEDIA 파일에 대하여 접근할 때 결정되는 값

`MEDIA_ROOT=os.path.join(BASE_DIR, 'media')`

실제로 파일이 저장될 위치

```
MEDIA_URL='/media/'
```

각 media 파일에 대한 URL의 고정값을 설정할 수 있음.

```
MEDIA_ROOT=os.path.join(BASE_DIR, 'media')
```

실제로 파일이 저장될 위치

프로젝트의 urls.py

```
instagram > 🗄️ urls.py > ...  
1  from django.contrib import admin  
2  from django.urls import path,include  
3  from django.conf.urls.static import static  
4  from django.conf import settings  
5  
6  urlpatterns = [  
7      path('admin/', admin.site.urls),  
8      path('',include('insta.urls')),  
9  ]  
10 |  
11 urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)  
12 urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

urls.py에 Media file을 제공(serving)하는 URL 패턴 등록 부분 수정
→올린 이미지를 읽어올 수 있음



ERROR

- insta
 - > __pycache__
 - > migrations
 - templates
 - insta
 - base.html
 - main.html
 - __init__.py
 - admin.py
 - apps.py
 - models.py 9+
 - tests.py
 - urls.py
 - views.py
- instagram
 - > __pycache__
 - __init__.py
 - settings.py
 - urls.py 9+
 - wsai.py
- media
 - DSC01655.JPG U
- db.sqlite3 M
- manage.py

media > DSC01655.JPG



`MEDIA_ROOT= os.path.join(BASE_DIR, 'media')`

다시 main.html 를 수정해봅시다.

```
<!-- post 추가-->
{% for post in posts %}
<div class="card mt-5">
  <div class="card-header">
    {{ post.user }}
    <span class="float-right" data-toggle="tooltip" data-placement="right"
      title="{{ post.updated | date:'Y년 n월 j일 a g:i' }}">{{ post.updated | timesince }}전</span></div>
    
    <div class="card-body" style="border-top: 0.5px solid ■ lightgray">
      <p class="card-text">{{ post.caption }}</p>
    </div>
    <div class="card-footer text-muted">
  </div>
<!-- post 끝-->
```

```
class User(AbstractUser): #여기는 () 상속을 받아온 부분
    1 profile=models.ImageField()

class Post(models.Model):
    1 user=models.ForeignKey(User,on_delete=models.CASCADE)
    3 image=models.ImageField()
    caption=models.TextField()
    created=models.DateTimeField(auto_now_add=True)
    2 updated=models.DateTimeField(auto_now=True)
```

```
def main(request):
    posts = Post.objects.all()
    sort = request.GET.get('sort', '')
```

url : media 폴더 부터 시작하는 상대경로

```
<!-- post 추가-->
```

```
{% for post in posts %}
```

```
<div class="card mt-5">
```

```
<div class="card-header">
```

```
{{ post.user }}
```

```
<span class="float-right" data-toggle="tooltip" data-placement="right"
```

```
title=2 {{ post.updated | date:'Y년 n월 j일 a g:i' }}">{{ post.updated | timesince }}전</span></div>
```

```

```

```
<div class="card-body" style="border-top: 0.5px solid lightgray">
```

```
<p class="card-text">{{ post.caption }}</p>
```

```
</div>
```

```
<div class="card-footer text-muted">
```

```
</div>
```

```
<!-- post 끝-->
```

날짜를 그 날짜 이후의 시간으로 포맷함

<https://docs.djangoproject.com/en/2.2/ref/templates/builtins/>

프로필 사진이 너무 큼!

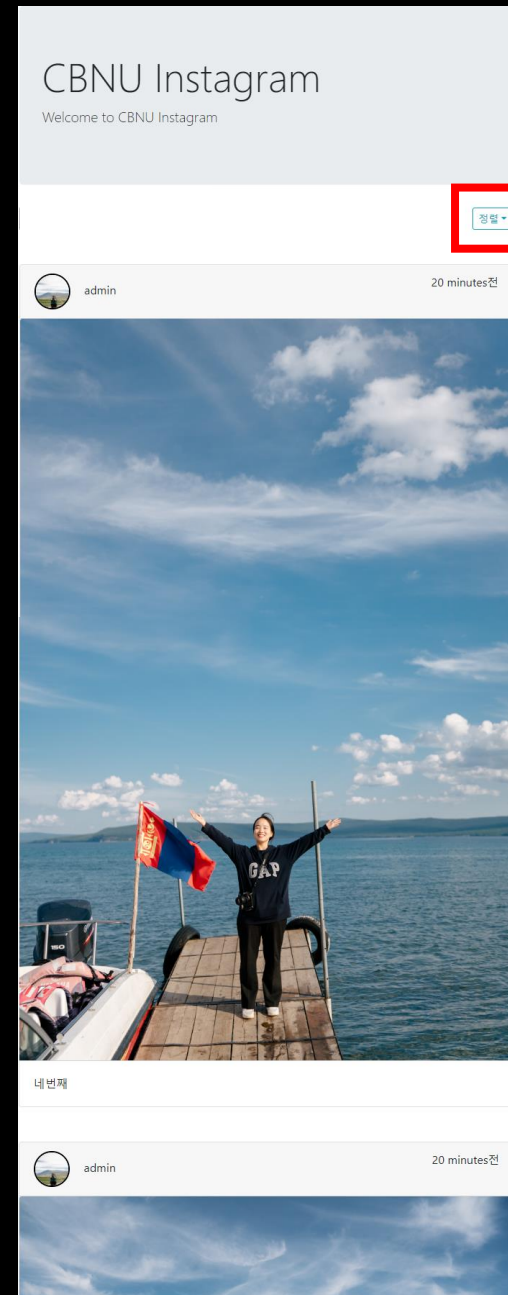
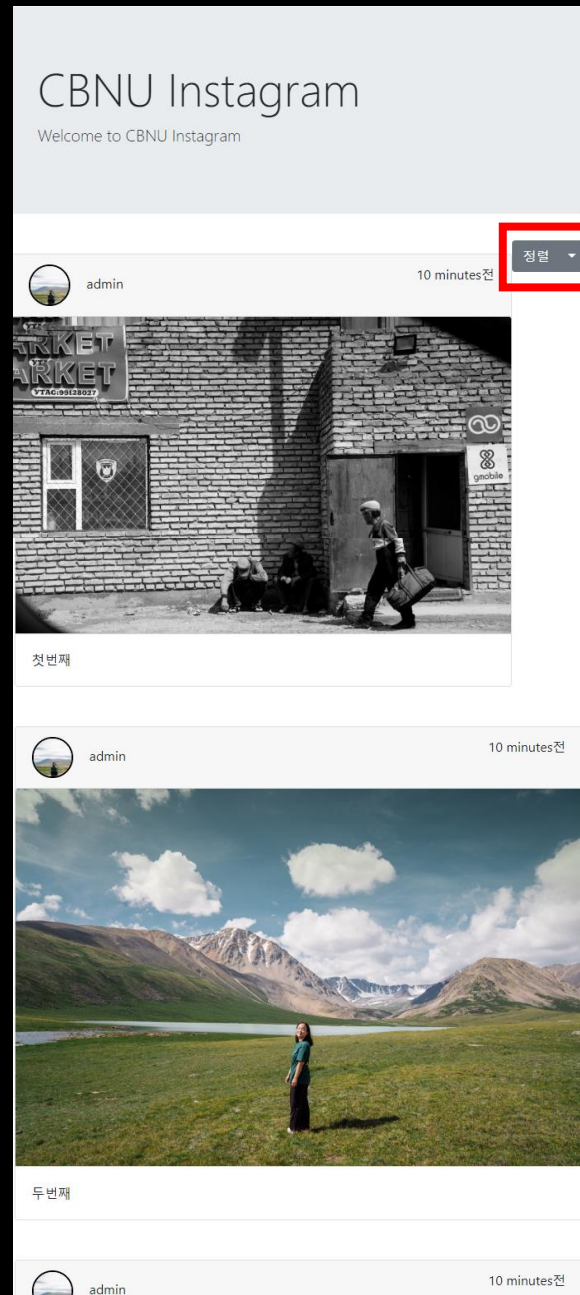
CSS를 수정

https://zetawiki.com/wiki/CSS_%EC%9B%90%ED%98%95_%EC%82%AC%EC%A7%84

```
<style>
  .profile{
    border: 2px solid black;
    border-radius: 50%;
    height: 50px;
    width: 50px;
  }
</style>

{% endfor %}
```

과제 1
제출기한
2019.9.15일
PM 11:59



이런식으로
정렬 드랍박스가
위로 가게 만들기

과제 2
제출기한
2019.9.15일
PM 11:59

좋아요 개수 낮은순으로 정렬해오기

만약 이런 오류가 뜬다면

```
ValueError at /
```

```
The 'profile' attribute has no file associated with it.
```

프로필 사진을 등록하지 않아서 뜨는 오류입니다.
admin 페이지로 가서 프로필 사진 등록해주세요