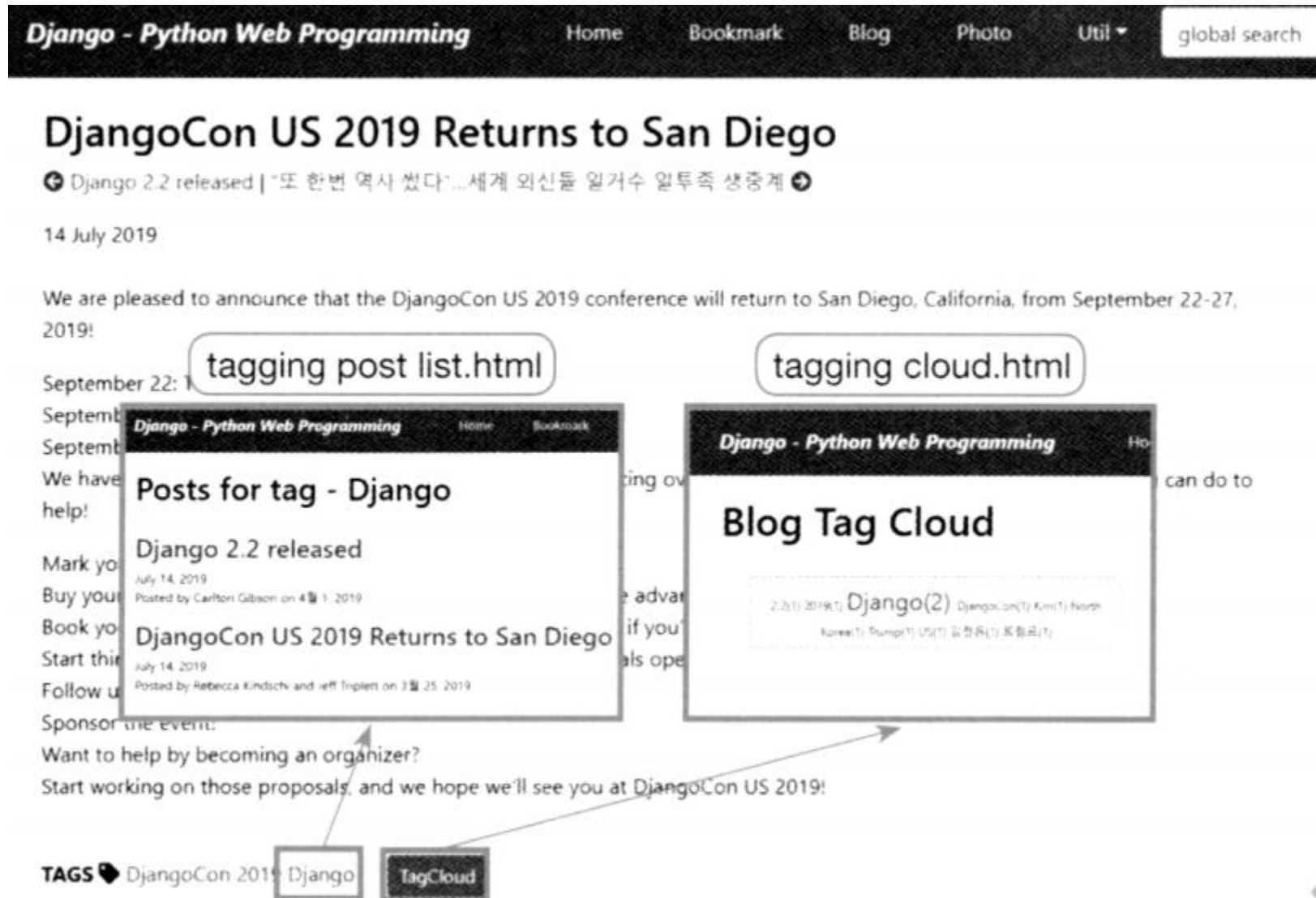

Blog 앱 확장

- Tag 달기 -

애플리케이션 설계하기

❖ 화면 설계



애플리케이션 설계하기

❖ 테이블 설계

○ Post 모델 클래스

필드명	타입	제약 조건	설명
id	Integer	PK, Auto Increment	기본 키
title	CharField(50)		포스트 제목
slug	SlugField(50)	Unique	포스트 제목 별칭
description	CharField(100)	Blank	포스트 내용 한 줄 설명
content	TextField		포스트 내용 기록
create_date	DateTimeField	auto_now_add	포스트를 생성한 날짜
modify_dt	DateTimeField	auto_now	포스트를 수정한 날짜
tags	TaggableManager	Blank, Null	포스트에 등록된 태그

애플리케이션 설계하기

❖ URL 설계

URL 패턴	뷰 이름	템플릿 파일명
/blog/	PostLV(ListView)	post_all.html
/blog/post/	PostLV(ListView)	post_all.html
/blog/post/django-example/	PostDV(DetailView)	post_detail.html
/blog/archive/	PostAV(ArchiveIndexView)	post_archive.html
/blog/archive/2012/	PostYAV(YearArchiveView)	post_archive_year.html
/blog/archive/2012/nov/	PostMAV(MonthArchiveView)	post_archive_month.html
/blog/archive/2012/nov/10/	PostDAV(DayArchiveView)	post_archive_day.html
/blog/archive/today/	PostTAV(TodayArchiveView)	post_archive_day.html
/blog/tag/	TagCloudTV(TemplateView)	taggit_cloud.html
/blog/tag/tagname/	TaggedObjectLV(ListView)	taggit_post_list.html

애플리케이션 설계하기

❖ 작업절차

작업 순서	관련 명령/파일	필요한 작업 내용
뼈대 만들기	startproject	(2장에서 이미 완료했으므로 생략)
	settings.py	
	migrate	taggit_post_list.html
	createsuperuser	
	startapp	(1장에서 django-taggit 등 두 패키지 설치함)
	settings.py	taggit, taggit_templatetags2 두 앱을 등록
모델 코딩하기	models.py	tags 필드 추가
	admin.py	태그 관련 내용 추가
	makemigrations	모델이 변경되므로 이를 데이터베이스에 반영
	migrate	
URLconf 코딩하기	urls.py	URL 정의 추가
뷰 코딩하기	views.py	뷰 로직 추가
템플릿 코딩하기	templates 디렉터리	템플릿 파일 추가
그 외 코딩하기	-	(없음)

개발 코딩하기

❖ 태깅 지원 패키지 설치

```
$ pip install django-taggit django-taggit-templatetags2
```

또는

```
$ conda install django-taggit django-taggit-templatetags2
```

django-taggit, django-taggit-templatetags2 공식 문서 사이트

<https://django-taggit.readthedocs.io/en/latest/index.html>

<https://github.com/fizista/django-taggit-templatetags2>

개발 코딩하기

❖ mysite/settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
  
    'taggit.apps.TaggitAppConfig',  
    'taggit_templatetags2',  
  
    'bookmark.apps.BookmarkConfig',  
    'blog.apps.BlogConfig',  
]  
  
:  
  
TAGGIT_CASE_INSENSITIVE = True  
TAGGIT_LIMIT = 50
```

개발 코딩하기

❖ blog/models.py

```
from django.db import models
from django.urls import reverse
```

```
from taggit.managers import TaggableManager
```

```
class Post(models.Model):
    title = models.CharField(verbose_name='TITLE', max_length=50)
    slug = models.SlugField('SLUG', unique=True, allow_unicode=True,
                            help_text='one word for title alias.')
    description = models.CharField('DESCRIPTION', max_length=100, blank=True,
                                   help_text='simple description text.')
    content = models.TextField('CONTENT')
    create_dt = models.DateTimeField('CREATE DATE', auto_now_add=True)
    modify_dt = models.DateTimeField('MODIFY DATE', auto_now=True)
    tags = TaggableManager(blank=True)

    :
```


개발 코딩하기

❖ blog/admin.py

```
from django.contrib import admin
from blog.models import Post

@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('id', 'title', 'modify_dt', 'tag_list')
    list_filter = ('modify_dt',)
    search_fields = ('title', 'content')
    prepopulated_fields = {'slug': ('title',)}

    def get_queryset(self, request):
        return super().get_queryset(request).prefetch_related('tags')

    def tag_list(self, obj):
        return ', '.join(o.name for o in obj.tags.all())
```

개발 코딩하기

❖ 데이터베이스에 모델 반영하기

```
$ python manage.py makemigrations blog  
$ python manage.py migrate
```

개발 코딩하기

❖ blog/views.py

```
from django.views.generic import ListView, DetailView, TemplateView
:

#--- Tag View
class TagCloudTV(TemplateView):
    template_name = 'taggit/taggit_cloud.html'

class TaggedObjectLV(ListView):
    template_name = 'taggit/taggit_post_list.html'
    model = Post

    def get_queryset(self):
        return Post.objects.filter(tags__name=self.kwargs.get('tag'))

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['tagname'] = self.kwargs['tag']
        return context
```

개발 코딩하기

❖ blog/urls.py

```
from django.urls import path, re_path
from blog import views

app_name = 'blog'
urlpatterns = [
    :
    # Example: /blog/tag/
    path('tag/', TagCloudTV.as_view(), name='tag_cloud'),

    # Example: /blog/tag/tagname/
    path('tag/<str:tag>', TaggedObjectLV.as_view(),
        name='tagged_object_list'),

]
```

개발 코딩하기

❖ blog/templates/blog/post_detail.html

```
{% extends "base.html" %}

{% block title %}post_detail.html{% endblock %}

{% block content %}
    <h2>{{ object.title }}</h2>

    <p>
        {% if object.get_next %}
        <a href="{{ object.get_next.get_absolute_url }}"
            title="View previous post">
            <i class="fas fa-arrow-circle-left"></i> {{ object.get_next }}
        </a>
        {% endif %}

        {% if object.get_previous %}
        | <a href="{{ object.get_previous.get_absolute_url }}"
            title="View next post">
            {{ object.get_previous }} <i class="fas fa-arrow-circle-right"></i>
        </a>
        {% endif %}
    </p>
```

개발 코딩하기

❖ blog/templates/blog/post_detail.html

```
<div>{{ object.modify_dt|date:"j F Y" }}</div>
<br>

<div>
    {{ object.content|linebreaks }}
</div>

<br>
<div>
    <b>TAGS</b> <i class="fas fa-tag"></i>
    {% load taggit_templatetags2_tags %}
    {% get_tags_for_object object as "tags" %}
    {% for tag in tags %}
    <a href="{% url 'blog:tagged_object_list' tag.name %}">{{tag.name}}</a>
    {% endfor %}
    &emsp;<a href="{% url 'blog:tag_cloud' %}">
        <span class="btn btn-info btn-sm">TagCloud</span> </a>
</div>
{% endblock %}
```

개발 코딩하기

❖ blog/templates/taggit/taggit_cloud.html

```
{% extends "base.html" %}

{% block title %}taggit_cloud.html{% endblock %}

{% block extra-style %}
<style type="text/css">
.tag-cloud {
    width: 40%;
    margin-left: 30px;
    text-align: center;
    padding: 5px;
    border: 1px solid orange;
    background-color: #ffc;
}
.tag-1 {font-size: 12px;}
.tag-2 {font-size: 14px;}
.tag-3 {font-size: 16px;}
.tag-4 {font-size: 18px;}
.tag-5 {font-size: 20px;}
.tag-6 {font-size: 24px;}
</style>
{% endblock %}
```

개발 코딩하기

❖ blog/templates/taggit/taggit_cloud.html

```
{% block content %}
<h1>Blog Tag Cloud</h1>
<br>

<div class="tag-cloud">
    {% load taggit_templatetags2_tags %}
    {% get_tagcloud as tags %}
    {% for tag in tags %}
        <span class="tag-{{tag.weight|floatformat:0}}">
            <a href="{% url 'blog:tagged_object_list' tag.name %}">
                {{tag.name}}({{tag.num_times}})</a>
            </span>
        {% endfor %}
    </div>
{% endblock %}
```


개발 코딩하기

❖ blog/templates/taggit/taggit_post_list.html

```
{% extends "base.html" %}

{% block title %}taggit_post_list.html{% endblock %}

{% block content %}

    <h1>Posts for tag - {{ tagname }}</h1>
    <br>

    {% for post in object_list %}
        <h2><a href='{{ post.get_absolute_url }}'>{{ post.title }}</a></h2>
        {{ post.modify_dt|date:"N d, Y" }}
        <p>{{ post.description }}</p>
    {% endfor %}

{% endblock %}
```