

## 3장. 포스트 수정/삭제 및 카테고리



# 포스트 삭제하기

[블로그홈](#) | [\(admin\)로그아웃](#) | [회원가입](#) | [글쓰기](#)

## girl

2021년 12월 26일 6:19 오후 (수정:2021년 12월 29일)

소녀의 뒷모습



127.0.0.1:8000 내용:

정말로 삭제하시겠습니까?

확인

취소

# 포스트 수정하기

## 1. 포스트 삭제 링크 만들기 – blog/post\_detail.html

```
<h2>{{ post.title }}</h2>
<h5>
    작성자 : {{ post.author }} (작성일 : {{ post.pub_date }})
    {% if request.user == post.author %}
        <a href="{% url 'blog:post_delete' post.id %}"
            onclick="return confirm('정말로 삭제하시겠습니까?')">삭제</a>
    {% endif %}
</h5>
<p>{{ post.content | linebreaks }}</p>
```

# 포스트 삭제하기

## 2. 포스트 삭제 URL 매핑하기 – blog/urls.py

```
urlpatterns = [  
    ...  
    path('post/delete/<int:pk>/', views.post_delete, name='post_delete'),  
]
```

# 포스트 삭제하기

## 3. post\_delete 함수 정의 – blog/views.py

```
@login_required(login_url='common:login')
def post_delete(request, post_id):
    # 포스트 삭제
    post = Post.objects.get(id=post_id)
    post.delete()
    return redirect('blog:index')
```

# 포스트 수정

[블로그홈](#) | [\(today\)로그아웃](#) | [회원가입](#) | [글쓰기](#)

## 인공지능 스피커

2021년 12월 29일 6:36 오후 (수정:2021년 12월 30일 4:49 오전) [수정](#) [삭제](#)

네이버 프렌즈 인공지능 스피커



# 포스트 수정하기

## 1. 포스트 수정 링크 만들기 – blog/post\_detail.html

```
<h5>
  작성자 : {{ post.author }} (작성일 : {{ post.pub_date }})
  {% if post.modify_date %}
    (수정일 : {{ post.modify_date }})
  {% endif %}
  <a href="{% url 'blog:index' %}">목록</a>
  {% if request.user == post.author %}
    <a href="{% url 'blog:post_modify' post.id %}">수정</a>
    <a href="{% url 'blog:post_delete' post.id %}"
      onclick="return confirm('정말로 삭제하시겠습니까?')">삭제</a>
  {% endif %}
</h5>
```

# 포스트 수정하기

## 2. 포스트 수정 URL 매핑하기 – blog/urls.py

```
urlpatterns = [  
    ....  
    path('post/modify/<int:pk>/', views.post_modify, name='post_modify'),  
]
```



# 포스트 수정 View

## 3. post\_modify 함수 정의 – blog/views.py

```
@login_required(login_url='common:login')
def post_modify(request, post_id):
    # 포스트 수정
    post = Post.objects.get(id=post_id)
    if request.method == "POST":
        form = PostForm(request.POST, request.FILES, instance=post)
        if form.is_valid():
            post = form.save(commit=False)
            post.modify_date = timezone.now()
            post.author = request.user
            post.save()
            return redirect('blog:detail', post_id=post_id)
        else:
            form = PostForm(instance=post)
    context = {'form': form}
    return render(request, 'blog/post_form.html', context)
```

# 포스트 수정 폼

## 4. 포스트 등록 폼에서 수정 – blog/post\_form.html

[블로그홈](#) | [\(today\)로그아웃](#) | [회원가입](#) | [글쓰기](#)

### 포스트 등록

제목:

내용: 

네이버 프렌즈  
인공지능 스피커

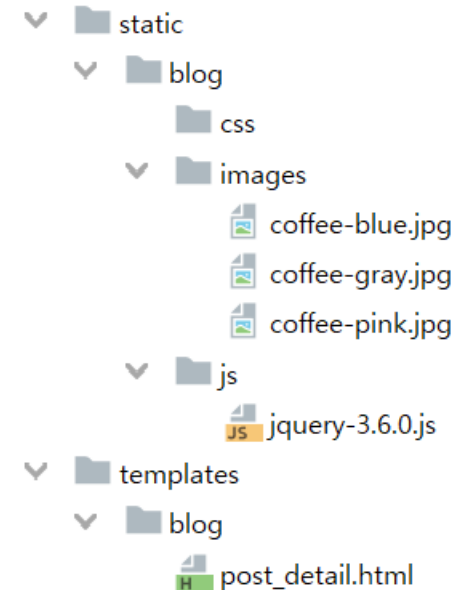
사진: 현재: <blog/images/2021/12/30/friends2.PNG> ☐ 취소

변경:  선택된 파일 없음

# 포스트 삭제하기

## 4. 제이쿼리로 삭제하기

```
<script src="{% static 'blog/js/jquery-3.6.0.js' %}"></script>
<script>
    $(function(){
        //포스트 삭제
        $(".delete").on('click', function(){
            if(confirm('정말로 삭제하시겠습니까?')){
                location.href = $(this).data('uri');
            }
        });
    });
</script>
```



# 카테고리(분류) 구현

[Home](#) [글쓰기](#) [\(today님\)로그아웃](#) [회원가입](#)

## Blog

### 자켓을 입은 Girl

작성자 : admin (작성일 : 2022년 3월 12일 9:29 오후)



라이프스타일

### 분류(Categories)

- 프로그래밍(2)
- 라이프스타일(1)
- 문화&예술(1)

# 카테고리 모델

## 1. Category 모델 만들기

```
# 카테고리 모델
class Category(models.Model):
    name = models.CharField(max_length=50, unique=True)
    slug = models.SlugField(max_length=200, unique=True, allow_unicode=True)

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return "/blog/category/{}".format(self.slug)
        # return f'/blog/category/{self.slug}'

    class Meta:
        verbose_name_plural = 'Categories'
```

# 카테고리 모델

## 1. Post 모델에 category 외래키 설정

```
class Post(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
    pub_date = models.DateTimeField()
    modify_date = models.DateTimeField(null=True, blank=True)
    photo = models.ImageField(upload_to='blog/images/%Y/%m/%d', null=True, blank=True)
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    category = models.ForeignKey(Category, null=True, blank=True, on_delete=models.SET_NULL)

    def __str__(self):
        return self.title
```

# 카테고리 URL

## 2. 카테고리 URL 매핑하기 – blog/urls.py

```
urlpatterns = [  
    ...  
    path('category/<str:slug>/', views.category_page, name='category_page'),  
]
```

# 카테고리 view

## 3. 카테고리 view – blog/views.py

```
def category_page(request, slug):  
    # 카테고리  
    category = Category.objects.get(slug=slug)  
    post_list = Post.objects.filter(category=category)  
    categories = Category.objects.all()  
    context = {  
        'category': category,  
        'post_list': post_list,  
        'categories': categories  
    }  
    return render(request, 'blog/post_list.html', context)
```



# 카테고리 view

## 3. 카테고리 view – index, detail 수정

```
def index(request):  
    # 포스트 목록  
    post_list = Post.objects.order_by('-pub_date')  
    categories = Category.objects.all()  
    context = {'post_list': post_list, 'categories': categories}  
    return render(request, 'blog/post_list.html', context)
```

```
def detail(request, post_id):  
    # 상세 페이지  
    post = Post.objects.get(id=post_id)  
    categories = Category.objects.all()  
    context = {'post': post, 'categories': categories}  
    return render(request, 'blog/post_detail.html', context)
```

# 카테고리 필드 폼에 추가하기

## 4. PostForm에 필드 추가 - category

```
class PostForm(forms.ModelForm):  
  
    class Meta:  
        model = Post  
        fields = ['title', 'content', 'photo', 'category']  
        labels = {  
            'title': '제목',  
            'content': '내용',  
            'photo': '사진',  
            'category': '카테고리'  
        }  
}
```

## 템플릿 – 카테고리 작성

### 5. 템플릿에 카테고리 작성 – post\_list.html, post\_detail.html

```
</section>
<aside>
  <h3>분류(Categories)</h3>
  <div>
    {% for category in categories %}
    <ul>
      <li>
        <a href="{{ category.get_absolute_url }}">
          {{ category }}({{ category.post_set.count }})
        </a>
      </li>
    </ul>
    {% endfor %}
  </div>
</aside>
```

# 템플릿 – 카테고리 작성

## 5. 템플릿에 카테고리 작성

```
{% block title %}Blog - 전체 보기{% endblock %}
{% block content %}
<section id="list">
  <h2>Blog</h2>
  {% if post_list %}
    {% for post in post_list %}
      <div>
        <hr>
        <h3><a href="{% url 'blog:detail' post.id %}">{{ post.title }}</a></h3>
        <!-- <h3><a href="blog/{{ post.id }}">{{ post.title }}</a></h3> -->
        {% if post.category %}
          <h4 class="category">{{ post.category }}</h4>
        {% endif %}
        <h5>작성자 : {{ post.author }} (작성일 : {{ post.pub_date }})</h5>
      </div>
    {% endfor %}
  {% else %}
    <p>No posts found.</p>
  {% endif %}
</section>
</div>
</div>
```

post\_list.html

# 템플릿 – 카테고리 작성

## 5. 템플릿에 카테고리 작성

```
<section id="detail">
  <h2>{{ post.title }}</h2>
  {% if post.category %}
  <h4 class="category">{{ post.category }}</h4>
  {% endif %}
  <h5>
    작성자 : {{ post.author }} (작성일 : {{ post.pub_date }})
    {% if post.modify_date %}
      (수정일 : {{ post.modify_date }})
    {% endif %}
    <a href="{% url 'blog:index' %}">목록</a>
    {% if request.user == post.author %}
      <a href="{% url 'blog:post_modify' post.id %}">수정</a>
```

post\_detail.html

# 카테고리 등록

## 6. 관리자 페이지에 등록 – blog/admin.py

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

admin.site.register(Post)

class CategoryAdmin(admin.ModelAdmin):
    prepopulated_fields = {'slug': ('name',)}

admin.site.register(Category, CategoryAdmin)
```

# 카테고리 등록 – 관리자 페이지

## 4. 관리자 페이지에 카테고리 추가

Start typing to filter...	category 추가
<b>BLOG</b>	
Categories + 추가	Name: 프로그래밍
Posts + 추가	Slug: 프로그래밍
<b>인증 및 권한</b>	
그룹 + 추가	
사용자(들) + 추가	

# 포스트 등록 – 관리자 페이지

## 4. 관리자 페이지에 포스트 등록 – 카테고리 선택

인공지능 스피커

Title:

인공지능 스피커

Content:

네이버 프렌즈  
AI 스피커 - 10만원 정도

Pub date:

날짜: 2022-03-12

오늘 | 📅

시각: 21:25:35

현재 | ⌚

Modify date:

날짜: 2022-03-13

오늘 | 📅

시각: 04:48:03

현재 | ⌚

Photo:

현재: blog/images/2022/03/12/friends2\_vNEagRy.PNG ☐ 취소

변경: 

파일 선택

 선택된 파일 없음

Author:

admin

✎ +

Category:

프로그래밍

✎ + ✖



# style.css

style.css

```
/* 메인 section */
#list, #detail{width:700px; float:left}
#list h3, #detail h2{float:left}
#list h4, #detail h4{float:right; margin-right: 30px; color: #00f;}
#list h5, #detail h5{clear:both;}

/* 카테고리 */
aside{width:200px; float:right; margin: 70px 20px 0 0;}
aside ul a{text-decoration:none; color:#222}
```

# 포스트 등록 폼

## ❖ 포스트 등록 폼에서 카테고리 추가하기

```
{% extends 'base.html' %}
{% block title %} Blog - 글쓰기 {% endblock %}
{% block content %}
<section>
  <h2>글쓰기</h2>
  <form action="" method="post" enctype="multipart/form-data">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">저장하기</button>
  </form>
</section>
{% endblock %}
```

# 카테고리 등록 – 포스트 등록 폼

## ❖ 상세 페이지에서 확인

[Home](#) [글쓰기](#) [\(today님\)로그아웃](#) [회원가입](#)

### 자켓을 입은 Girl

라이프스타일

작성자 : admin (작성일 : 2022년 3월 12일 9:29 오후) (수정일 : 2022년 3월 13일 4:17 오전)

---




#### 분류(Categories)

- 프로그래밍(2)
- 라이프스타일(1)
- 문화&예술(1)

# 깃으로 버전(소스) 관리하기

## ➤ 깃에 업로드하기

🔑 master ▾ 🔑 1 branch 🔖 0 tags Go to file Add file ▾ Code ▾

 **sugu2100** 포스트 수정, 삭제

f91aa39 2 minutes ago ⌚ 3 commits

📁 blog	포스트 수정, 삭제	2 minutes ago
📁 common	common 앱 생성	3 days ago
📁 media/blog/images/2021/12	포스트 수정, 삭제	2 minutes ago
📁 myblog	포스트 수정, 삭제	2 minutes ago
📁 static/blog	포스트 수정, 삭제	2 minutes ago
📁 templates	포스트 수정, 삭제	2 minutes ago
📄 .gitignore	common 앱 생성	3 days ago
📄 manage.py	blog site start	4 days ago