

① blog – urls.py 코딩 추가

```
1 from django.urls import path, re_path
2 from blog import views
3 from blog.views import *
4
5 app_name = 'blog' #네임스페이스
6 urlpatterns = [
7     path('', views.PostLV.as_view(), name='index'),
8     path('post/', views.PostLV.as_view(), name='post_list'),
9     re_path(r'^post/(?P<slug>[-\w]+)/$', views.PostDV.as_view(), name='post_detail'),
10    path('archive/', views.PostAV.as_view(), name='post_archive'),
11    path('archive/<int:year>/', views.PostYAV.as_view(), name='post_year_archive'),
12    path('archive/<int:year>/<str:month>/', views.PostMAV.as_view(), name='post_month_archive'),
13    path('archive/<int:year>/<str:month>/<int:day>/', views.PostDAV.as_view(), name='post_day_archive'),
14    path('archive/today/', views.PostTAV.as_view(), name='post_today_archive'),
15    path('tag/', views.TagCloudTV.as_view(), name='tag_cloud'),
16    path('tag/<str:tag>/', views.TaggedObjectLV.as_view(), name='tagged_object_list'),
17    path('search/', views.SearchFormView.as_view(), name='search'),
18 ]
```

② blog – forms 모듈 만들고 코딩 추가

```
from django import forms

class PostSearchForm(forms.Form):
    search_word = forms.CharField(label='찾는단어')
```

③ blog – views.py에 코딩 추가

- 임포트 추가

```
from django.views.generic import ListView, DetailView, TemplateView, FormView
from blog.forms import PostSearchForm
from django.db.models import Q #검색기능을 가지고 있는 클래스(색인)
from django.shortcuts import render #단축함수
```

- SearchFormView 클래스 만들어주기

```
#FormView
class SearchFormView(FormView):
    form_class = PostSearchForm
    template_name = 'blog/post_search.html'

    def form_valid(self, form):
        searchWord = form.cleaned_data['search_word']
        post_list = Post.objects.filter(
            Q(title__icontains=searchWord) |
            Q(description__icontains=searchWord) |
            Q(content__icontains=searchWord)
        ).distinct() #중복되는 것은 배제(dintinct)
        context = {}
        context['form'] = form
        context['search_term'] = searchWord
        context['object_list'] = post_list

        return render(self.request, self.template_name, context) #리다이렉트가 처리되지 않는다
```

④ base.html 수정

```
<li class="nav-item dropdown mx-1 btn btn-dark">
  <a class="nav-link dropdown-toggle text-white" href="#" data-toggle="dropdown">Util</a>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="{% url 'admin:index' %}">Admin</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="{% url 'blog:post_archive' %}">Archive</a>
    <a class="dropdown-item" href="{% url 'blog:search' %}">Search</a>
  </div>
</li>
```

⑤ blog – templates – blog - post_search.html 작성

```
{% extends "base.html" %}
{% block title %}포스트 찾기{% endblock %}

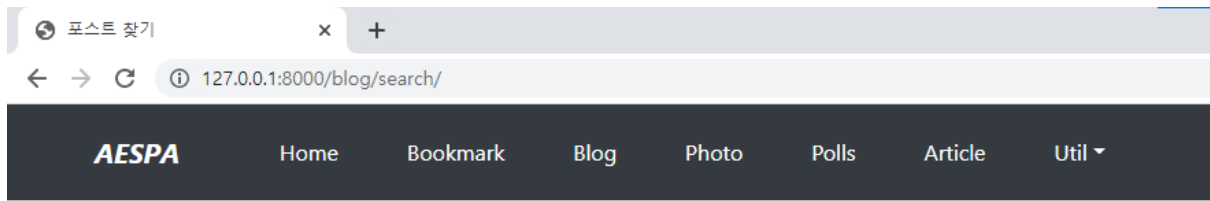
{% block content %}
<h1>블로그 찾기</h1>
<hr>
<form action='.' method="post">
{% csrf_token %} <!-- form 첫줄에 항상 넣어줘야함 -->
{{form.as_table}} <!-- 테이블 형태로 보여줌 as_table -->
<input type="submit" value="찾기" class="btn btn-primary btm-sm">
</form>

<br/><br/>
{% if object_list %}
  {% for post in object_list %}
    <h2><a href='{{post.get_absolute_url}}'> <!-- 절대경로를 가져오기 -->
      {{post.title}}
    </a></h2>
    {{post.modify_date|date:"N d, Y"}}
    <p>{{post.description}}</p>
  {% endfor %}

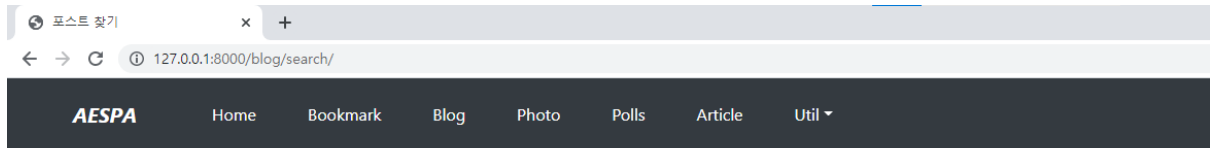
{% elif search_term %}
  <b><i>Search Word({{search_term}})Not Found !</i></b>
{% endif %}

{% endblock %}
```

▶▶▶ 결과 ◀◀◀



찾는단어:



찾는단어:

블랙맘바
카리나