

Update & Delete / Template Inheritance / Bootstrap

CRUD

CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 Create(생성), Read(읽기), Update(갱신), Delete(삭제)를 묶어서 일컫는 말



urls.py



```
1  from django.urls import path
2  from . import views
3
4  app_name = 'posts'
5  urlpatterns = [
6      path('', views.index, name="index"),
7      path('<int:post_id>/', views.detail, name='detail'),
8      path('new/', views.new, name='new'),
9      path('create/', views.create, name='create'),
10     path('<int:post_id>/edit/', views.edit, name='edit'),
11     path('<int:post_id>/update/', views.update, name='update'),
12 ]
```

```
def edit(request, post_id):  
    post = Post.objects.get(id = post_id)  
    context = {'post' : post }  
    return render(request, 'posts/edit.html', context)
```

```
def update(request, post_id):  
    post = Post.objects.get(id = post_id)  
    post.author = request.POST['author']  
    post.body = request.POST['body']  
    post.save()  
  
    return redirect('posts:detail', post_id = post.id)
```

```
{% load static %}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>GSM</title>
```

```
    <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
```

```
  </head>
```

```
  <body>
```

```
    <h1>Edit Post</h1>
```

```
    <form action="{% url 'posts:update' post.id %}" method="POST">
```

```
      {% csrf_token %}
```

```
      <h2>작 성 자</h2>
```

```
      <input type="text" name="author" value='{{ post.author }}' />
```

```
      <h2>본 문</h2>
```

```
      <textarea cols=40 rows=10 name="body"> {{post.body}} </textarea>
```

```
      <br>
```

```
      <input type="submit" value="수 정 하 기" />
```

```
    </form>
```

```
    <a href="{% url 'posts:index' %}">목 록</a>
```

```
  </body>
```

```
</html>
```

```
def edit(request, post_id):  
    post = Post.objects.get(id = post_id)  
    context = {'post' : post }  
    return render(request, 'posts/edit.html', context)
```

```
def update(request, post_id):  
    post = Post.objects.get(id = post_id)  
    post.author = request.POST['author']  
    post.body = request.POST['body']  
    post.save()  
  
    return redirect('posts:detail', post_id = post.id)
```

```
{% load static %}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>GSM Board</title>
```

```
    <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
```

```
  </head>
```

```
  <body>
```

```
    <h1>Post Detail</h1>
```

```
    {% if post %}
```

```
      <h2>작성자</h2>
```

```
      <p>{{ post.author }}</p>
```

```
      <h2>본 문</h2>
```

```
      <p>{{ post.body }}</p>
```

```
      <h2>게 시 일</h2>
```

```
      <p>{{ post.created_at }}</p>
```

```
    {% else %}
```

```
      <p>No posts</p>
```

```
    {% endif %}
```

```
    <a href="{% url 'posts:index' %}">문 투</a>
```

```
    <a href="{% url 'posts:edit' post.id %}">수 정 하 기</a>
```

```
  </body>
```

```
</html>
```

```
urls.py
from django.urls import path
from . import views

app_name = 'posts'
urlpatterns = [
    path('', views.index, name="index"),
    path('<int:post_id>/', views.detail, name='detail'),
    path('new/', views.new, name='new'),
    path('create/', views.create, name='create'),
    path('<int:post_id>/edit/', views.edit, name='edit'),
    path('<int:post_id>/update/', views.update, name='update'),
]

views.py
from django.shortcuts import render, redirect
from .models import Post
from django.utils import timezone

def index(request):
    posts = Post.objects.all()
    context = {'posts': posts}
    return render(request, 'posts/index.html', context)

def detail(request, post_id):
    post = Post.objects.get(id = post_id)
    context = {'post': post }
    return render(request, 'posts/detail.html', context)

def new(request):
    return render(request, 'posts/new.html')

def create(request):
    author = request.POST['author']
    body = request.POST['body']
    post = Post(author=author, body=body, created_at=timezone.now())
    post.save()

    return redirect('posts:detail', post_id = post.id)

def edit(request, post_id):
    post = Post.objects.get(id = post_id)
    context = [{'post': post}]
    return render(request, 'posts/edit.html', context)

def update(request, post_id):
    post = Post.objects.get(id = post_id)
    post.author = request.POST['author']
    post.body = request.POST['body']
    post.save()

    return redirect('posts:detail', post_id = post.id)
```

```
edit.html
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <title>GSM</title>
7         <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
8     </head>
9     <body>
10        <h1>Edit Post</h1>
11
12        <form action="{% url 'posts:update' post.id %}" method="POST">
13            {% csrf_token %}
14
15            <h2>작 성 자</h2>
16            <input type="text" name="author" value="{{ post.author }}" />
17
18            <h2>본 문</h2>
19            <textarea cols=40 rows=10 name="body"> {{post.body}}</textarea>
20
21            <br>
22            <input type="submit" value="수 정 하 기"/>
23        </form>
24
25        <a href="{% url 'posts:index' %}">목 록</a>
26    </body>
27 </html>
28
29
30
31
```


Post Detail

작성자

minyoung

본 문

나의 Django Project :D 가능 ?

게 시 일

2020년 10월 12일 9:52 오전

[목록 수정 하기](#)

Edit Post

작 성 자

본 문

나의 Django Project :D 가능 ?

[목록](#)

Edit Post

작 성 자

minyoung

본 문

나의 Django Project :D
수정하기를 시작합니다.

수정하기

[목록](#)

Post Detail

작성자

minyoung

본 문

나의 Django Project :D 수정하기를 시작합니다.

게 시 일

2020년 10월 12일 9:52 오전

[목록 수정하기](#)

Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
5	gsm	gsm_lunch	2020년 11월 3일 10:58 오전	보기
6	CODING	ZAM	2020년 11월 3일 11:02 오전	보기
7	애들아	장고를 다 까먹은건 아니지..?	2020년 11월 3일 11:23 오전	보기

[글 쓰기](#)

```
urls.py
from django.urls import path
from . import views

app_name = 'posts'
urlpatterns = [
    path('', views.index, name="index"),
    path('<int:post_id>/', views.detail, name='detail'),
    path('new/', views.new, name='new'),
    path('create/', views.create, name='create'),
    path('<int:post_id>/edit/', views.edit, name='edit'),
    path('<int:post_id>/update/', views.update, name='update'),
    path('<int:post_id>/delete/', views.delete, name='delete'),
]

15:1 Python Spaces: 4
```

```
views.py

return render(request, 'posts/edit.html', context)

def update(request, post_id):
    post = Post.objects.get(id = post_id)
    post.author = request.POST['author']
    post.body = request.POST['body']
    post.save()

    return redirect('posts:detail', post_id = post.id)

def delete(request, post_id):
    post = Post.objects.get(id = post_id)
    post.delete()

    return redirect('posts:index')
```

```
detail.html

1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <title>GSM Board</title>
7         <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
8     </head>
9     <body>
10         <h1>Post Detail</h1>
11         {% if post %}
12             <h2>작성자</h2>
13             <p>{{ post.author }}</p>
14
15             <h2>본 문</h2>
16             <p>{{ post.body }}</p>
17
18             <h2>게 시 일</h2>
19             <p>{{ post.created_at }}</p>
20         {% else %}
21             <p>No posts</p>
22         {% endif %}
23
24         <a href="{% url 'posts:index' %}">목 록</a>
25         <a href="{% url 'posts:edit' post.id %}">수 정 하 기</a>
26         <a href="{% url 'posts:delete' post.id %}">삭 제 하 기</a>
27     </body>
28 </html>
29
30
```

Post Detail

작성자

애들아

본 문

장고를 다 까먹은건 아니지..?

게 시 일









2020년 11월 3일 11:23 오전

[목록](#) [수정하기](#) [삭제하기](#)

Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING	ZAM	2020년 11월 3일 11:02 오전	보기

[글 쓰기](#)

- ▼  posts
 -  migrations
 - ▼  templates
 - ▼  posts
 -  detail.html
 -  edit.html
 -  index.html
 -  new.html

index.html

x



```
{% load static %}
<!DOCTYPE html>
<html>
  <head>
    <title>GSM</title>
  </head>
  <body>
    <h1>Posts</h1>
    <!--<ul>
      <li>나의 첫 Django project</li>-->
    {% if posts %}
      <table>
        <thead>
          <tr>
            <th>#</th>
            <th>작성 자</th>
            <th>본문</th>
            <th>게시일</th>
            <th>상세</th>
          </tr>
        </thead>
        <tbody>
          {% for post in posts %}
            <tr>
              <th>{{ post.id }}</th>
              <td>{{ post.author }}</td>
              <td>{{ post.body }}</td>
              <td>{{ post.created_at }}</td>
              <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    {% else %}
      <p>No Posts.</p>
    {% endif %}

    <a href="{% url 'posts:new' %}">글 쓰 기</a>
  </body>
</html>
```



detail.html

x



```
1  {% load static %}
2
3  <!DOCTYPE html>
4  <html>
5    <head>
6      <title>GSM Board</title>
7      <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
8    </head>
9    <body>
10     <h1>Post Detail</h1>
11     {% if post %}
12       <h2>작성 자</h2>
13       <p>{{ post.author }}</p>
14
15       <h2>본 문</h2>
16       <p>{{ post.body }}</p>
17
18       <h2>게 시 일</h2>
19       <p>{{ post.created_at }}</p>
20     {% else %}
21       <p>No posts</p>
22     {% endif %}
23
24     <a href="{% url 'posts:index' %}">목 록</a>
25     <a href="{% url 'posts:edit' post.id %}">수 정 하 기</a>
26     <a href="{% url 'posts:delete' post.id %}">삭 제 하 기</a>
27   </body>
28 </html>
29
30
```

```

index.html x +
{% load static %}
<!DOCTYPE html>
<html>
  <head>
    <title>GSM</title>
  </head>
  <body>
    <h1>Posts</h1>
    <!--<ul>
      <li>나의 첫 Django project</li>-->
    {% if posts %}
      <table>
        <thead>
          <tr>
            <th>#</th>
            <th>작성 자</th>
            <th>본문</th>
            <th>게시일</th>
            <th>상세</th>
          </tr>
        </thead>
        <tbody>
          {% for post in posts %}
            <tr>
              <th>{{ post.id }}</th>
              <td>{{ post.author }}</td>
              <td>{{ post.body }}</td>
              <td>{{ post.created_at }}</td>
              <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    {% else %}
      <p>No Posts.</p>
    {% endif %}

    <a href="{% url 'posts:new' %}">글 쓰 기</a>
  </body>
</html>

```

```

detail.html x +
{% load static %}

<!DOCTYPE html>
<html>
  <head>
    <title>GSM Board</title>
    <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
  </head>
  <body>
    <h1>Post Detail</h1>
    {% if post %}
      <h2>작성 자</h2>
      <p>{{ post.author }}</p>

      <h2>본 문</h2>
      <p>{{ post.body }}</p>

      <h2>게 시 일</h2>
      <p>{{ post.created_at }}</p>
    {% else %}
      <p>No posts</p>
    {% endif %}

    <a href="{% url 'posts:index' %}">목 록</a>
    <a href="{% url 'posts:edit' post.id %}">수 정 하 기</a>
    <a href="{% url 'posts:delete' post.id %}">삭 제 하 기</a>
  </body>
</html>

```

Template Inheritance

gsm_Django.com



gsm_Django.com 의 모든 페이지에는

1. 공통적인 Style을 사용하고 있다.
2. 화면 상단에 Navigation이 있다.
3. 하단 Footer에는 회사 정보를 담는다.

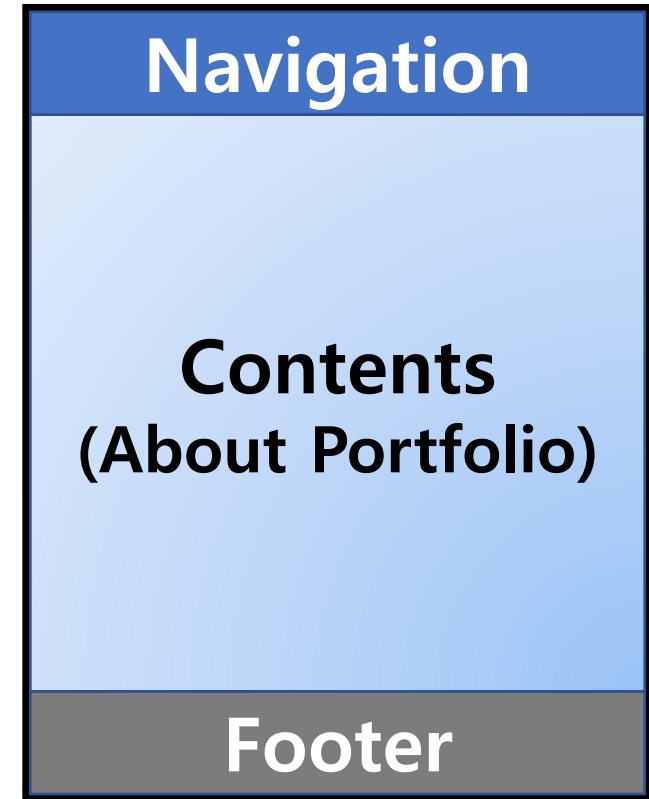
gsm_Django.com/posts

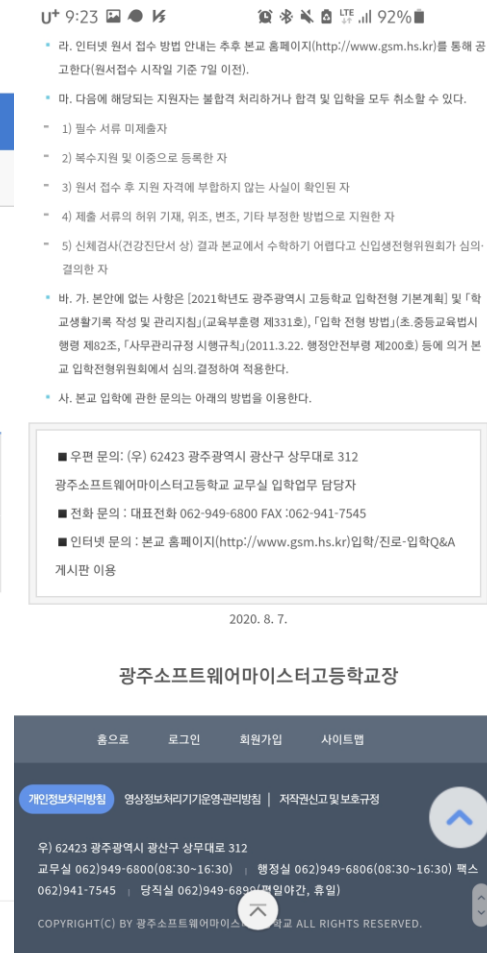
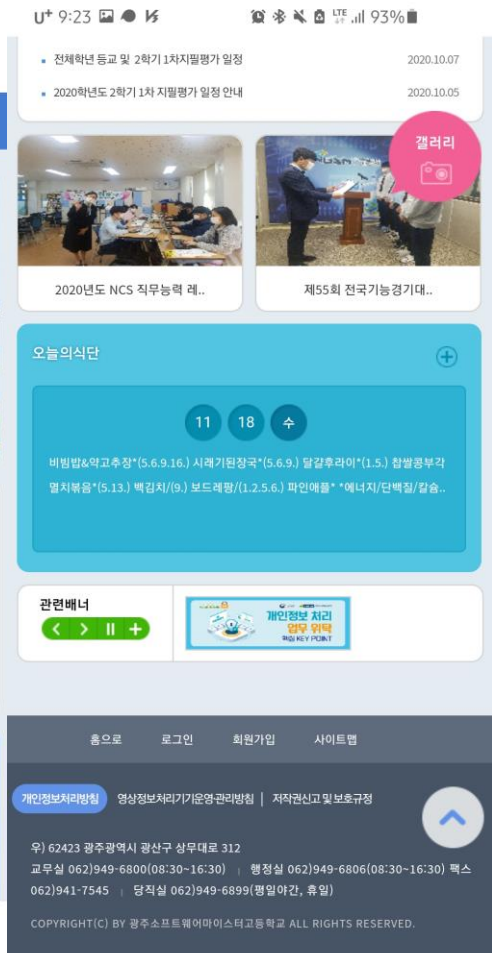


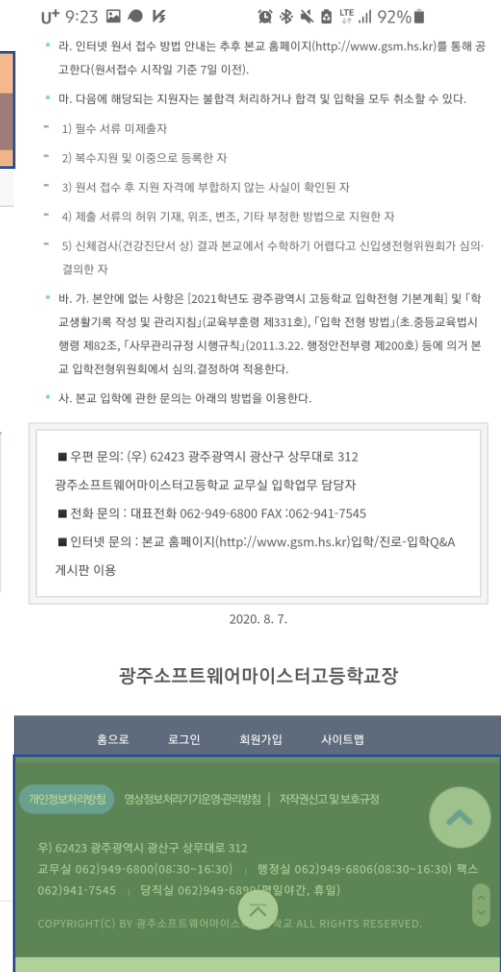
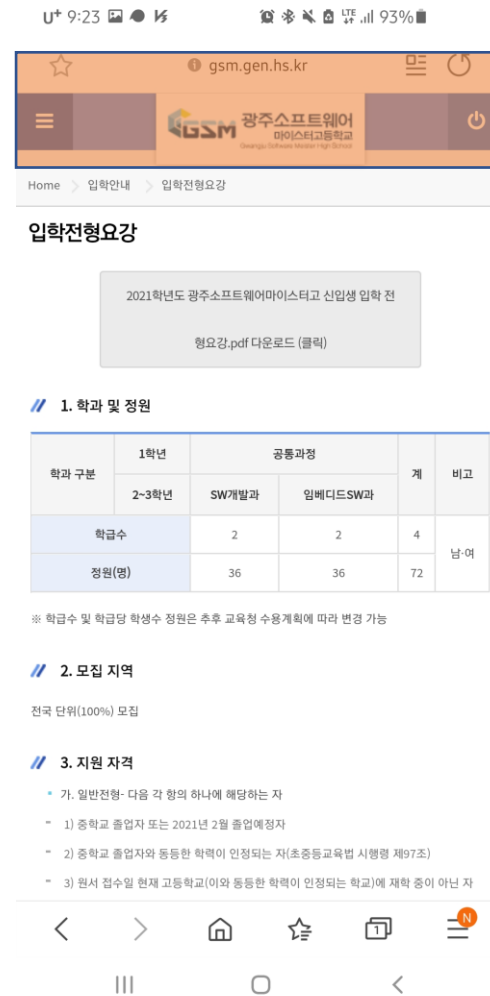
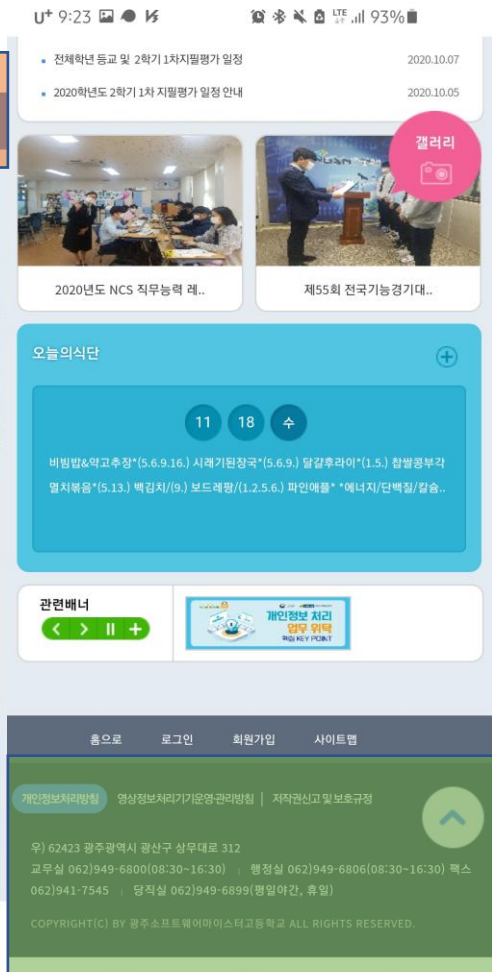
gsm_Django.com/blogs



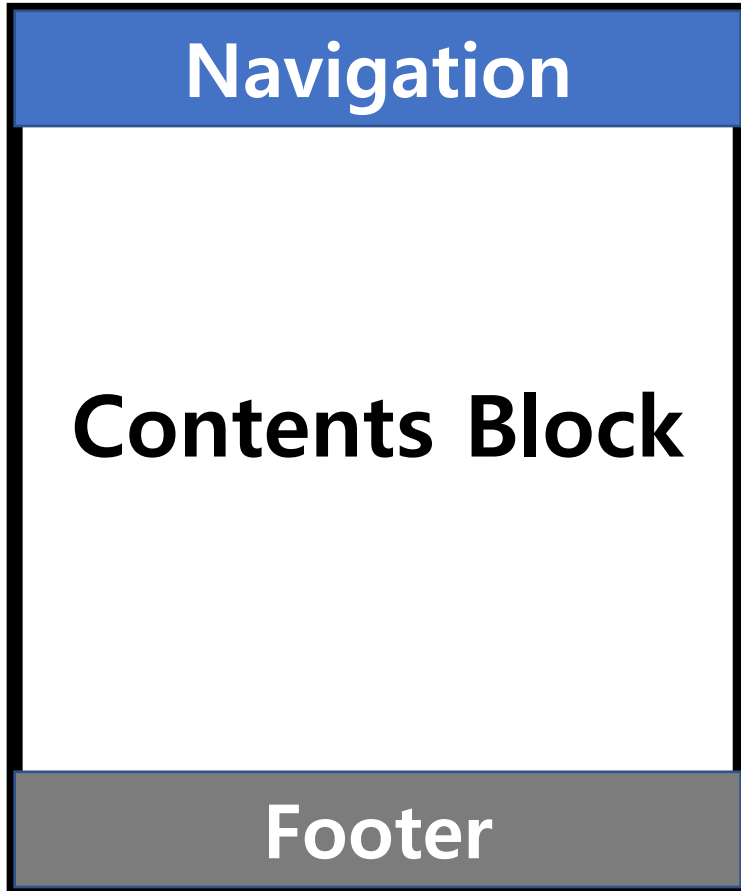
[gsm_Django.com/ Portfolios](http://gsm_Django.com/Portfolios)







base.html



모든 웹페이지에서 공통적으로 사용하는 Base Template

페이지가 달라져도 이 Contents Block 에 들어갈 내용만 바꾸면 된다.

base.html

Navigation

Contents Block

Footer

index.html

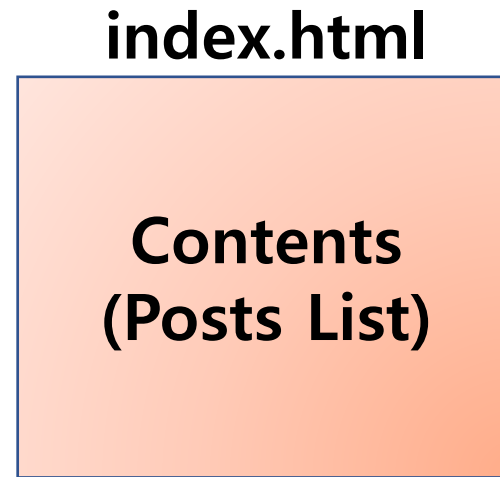
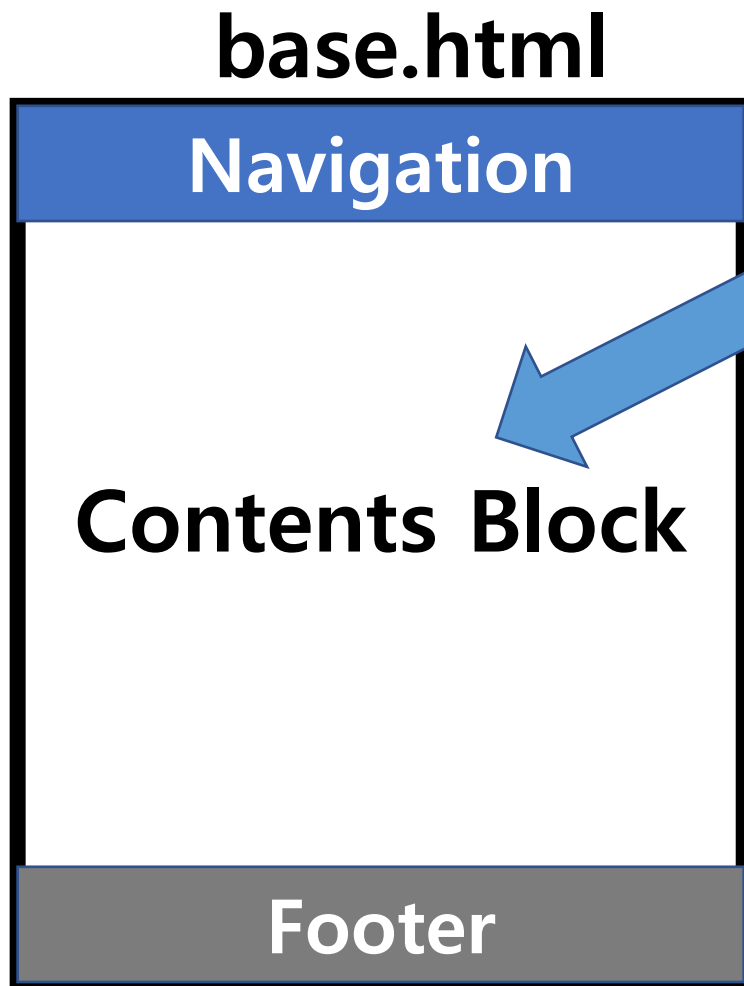
**Contents
(Posts List)**

new.html

**Contents
(New Post)**

detail.html

**Contents
(Posts Detail)**





Go to Anything (Ctrl-P)



gsm_django2 - /h

gsm

gsm

posts

static

style.css

templates

base.html

db.sqlite3

manage.py

exam.html

README.md



base.html



```
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <title>GSM</title>
7         <link rel="stylesheet" href="{% static 'style.css' %}" type="text/css" />
8         {% block style %}
9         {% endblock %}
10    </head>
11    <body>
12        {% block content %}
13        {% endblock %}
14    </body>
15 </html>
```



Go to Anything (Ctrl-P)



gsm_django2 - /h

gsm

gsm

posts

static

style.css

templates

base.html

db.sqlite3

manage.py

exam.html

README.md



base.html



```
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <title>GSM</title>
7     <link rel="stylesheet" href="{% static 'style.css' %}" type="text/css" />
8     {% block style %}
9     {% endblock %}
10  </head>
11  <body>
12    {% block content %}
13    {% endblock %}
14  </body>
15 </html>
```



Go to Anything (Ctrl-P)

- gsm_django2 - /h
 - gsm
 - posts
 - static
 - style.css
 - templates
 - base.html
 - db.sqlite3
 - manage.py
 - exam.html
 - README.md

```
base.html
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <title>GSM</title>
7     <link rel="stylesheet" href="{% static 'style.css' %}" type="text/css" />
8     {% block style %}
9     {% endblock %}
10  </head>
11  <body>
12    {% block content %}
13    {% endblock %}
14  </body>
15 </html>
```

```
style.css
1 * {
2   margin: 0px;
3   padding: 0px;
4   box-sizing: border-box;
5 }
6
```



Go to Anything (Ctrl-P)

gsm_django2 - /h

- gsm
 - gsm
 - posts
 - static
 - style.css
 - templates
 - base.html
- db.sqlite3
- manage.py
- exam.html
- README.md

base.html

```
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <title>GSM</title>
7     <link rel="stylesheet" href="{% static 'style.css' %}" type="text/css" />
8     {% block style %}
9     {% endblock %}
10  </head>
11  <body>
12    {% block content %}
13    {% endblock %}
14  </body>
15 </html>
16
17
```

style.css

```
1 * {
2   margin: 0px;
3   padding: 0px;
4   box-sizing: border-box;
5 }
6
```

Go to Anything (Ctrl-P)



settings.py



asm_diano2 - /home/ubuntu/enviro

gsm

gsm

__init__.py

asgi.py

settings.py

urls.py

wsgi.py

posts

static

style.css

templates

base.html

db.sqlite3

manage.py

exam.html

README.md

```
13 from pathlib import Path
14 import os
```

```
53
54
55
56 TEMPLATES = [
57     {
58         'BACKEND': 'django.template.backends.django.DjangoTemplates',
59         'DIRS': [os.path.join(BASE_DIR, 'templates')],
60         'APP_DIRS': True,
61         'OPTIONS': {
62             'context_processors': [
63                 'django.template.context_processors.debug',
64                 'django.template.context_processors.request',
65                 'django.contrib.auth.context_processors.auth',
66                 'django.contrib.messages.context_processors.messages',
67             ],
68         },
69     ],
70 ]
```

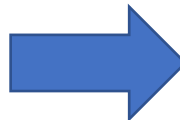
```
119 # Static files (CSS, JavaScript, Images)
120 # https://docs.djangoproject.com/en/3.1/howto/static-files/
121
122 STATIC_URL = '/static/'
123
124 STATICFILES_DIRS = [
125     os.path.join(BASE_DIR, 'static'),
126 ]
```

```
78 DATABASES = {
79     'default': {
```

```
{% load static %}
<!DOCTYPE html>
<html>
  <head>
    <title>GSM</title>
  </head>
  <body>
    <h1>Posts</h1>
    <!--<ul>
      <li>나의 첫 Django project</li>-->
    {% if posts %}
      <table>
        <thead>
          <tr>
            <th>#</th>
            <th>작성자</th>
            <th>문문</th>
            <th>게시일</th>
            <th>상세</th>
          </tr>
        </thead>
        <tbody>
          {% for post in posts %}
            <tr>
              <th>{{ post.id }}</th>
              <td>{{ post.author }}</td>
              <td>{{ post.body }}</td>
              <td>{{ post.created_at }}</td>
              <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    {% else %}
      <p>No Posts.</p>
    {% endif %}

    <a href="{% url 'posts:new' %}">글 쓰기</a>
  </body>
</html>
```

기존 index.html



```
index.html x (+)

{% extends "base.html" %}

{% load static %}

{% block style %}
  <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
{% endblock %}

{% block content %}
  <h1>Posts</h1>

  {% if posts %}
    <table>
      <thead>
        <tr>
          <th>#</th>
          <th>작성자</th>
          <th>문문</th>
          <th>게시일</th>
          <th>상세</th>
        </tr>
      </thead>
      <tbody>
        {% for post in posts %}
          <tr>
            <th>{{ post.id }}</th>
            <td>{{ post.author }}</td>
            <td>{{ post.body }}</td>
            <td>{{ post.created_at }}</td>
            <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
          </tr>
        {% endfor %}
      </tbody>
    </table>
  {% else %}
    <p>No Posts.</p>
  {% endif %}

  <a href="{% url 'posts:new' %}">글 쓰기</a>
{% endblock %}
```

수정 index.html
(Template Inheritance 적용)

```
base.html
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <title>GSM</title>
7         <link rel="stylesheet" href="{% static 'style.css' %}" type="text/css" />
8         {% block style %}
9         {% endblock %}
10    </head>
11    <body>
12        {% block content %}
13        {% endblock %}
14    </body>
15 </html>
```

base.html

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

{% load static %}

{% block style %}
    <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
{% endblock %}

{% block content %}
    <h1>Posts</h1>

    {% if posts %}
        <table>
            <thead>
                <tr>
                    <th>#</th>
                    <th>작성자</th>
                    <th>문명</th>
                    <th>게시일</th>
                    <th>상세</th>
                </tr>
            </thead>
            <tbody>
                {% for post in posts %}
                    <tr>
                        <th>{{ post.id }}</th>
                        <td>{{ post.author }}</td>
                        <td>{{ post.body }}</td>
                        <td>{{ post.created_at }}</td>
                        <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
                    </tr>
                {% endfor %}
            </tbody>
        </table>
    {% else %}
        <p>No Posts.</p>
    {% endif %}

    <a href="{% url 'posts:new' %}">글 쓰기</a>
    {% endblock %}
```

수정 index.html
(Template Inheritance 적용)

```
base.html
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <title>G</title>
7     <link rel="stylesheet" href="{% static 'style.css' %}" type="text/css" />
8     {% block style %}
9     {% endblock %}
10  </head>
11  <body>
12    {% block content %}
13    {% endblock %}
14  </body>
15 </html>
```

base.html

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

{% load static %}

{% block style %}
  <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
{% endblock %}

{% block content %}
  <h1>Posts</h1>

  {% if posts %}
    <table>
      <thead>
        <tr>
          <th>#</th>
          <th>작성자</th>
          <th>문</th>
          <th>게시일</th>
          <th>상세</th>
        </tr>
      </thead>
      <tbody>
        {% for post in posts %}
          <tr>
            <th>{{ post.id }}</th>
            <td>{{ post.author }}</td>
            <td>{{ post.body }}</td>
            <td>{{ post.created_at }}</td>
            <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
          </tr>
        {% endfor %}
      </tbody>
    </table>
  {% else %}
    <p>No Posts.</p>
  {% endif %}

  <a href="{% url 'posts:new' %}">글 쓰 기</a>
{% endblock %}
```

수정 index.html
(Template Inheritance 적용)

GSM

x +

← → ↺

7499bd749db844dbb360c7bcc6b805e1.vfs.cloud9.us-east-1.amazonaws.com/posts/

☆ 📄 ⚙️ 👤 민영 ⋮

Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING ZAM		2020년 11월 3일 11:02 오전	보기

[글 쓰기](#)

🔍 📄 | Elements Console Sources Network >> 2 ⚙️ ⋮ ✕

```
<!DOCTYPE html>
<html>
  <head>
    <title>GSM</title>
    <link rel="stylesheet" href="/static/style.css" type="text/css">
    <link rel="stylesheet" href="/static/posts/style.css" type="text/css">
  </head>
  <body> == $0
    <h1>Posts</h1>
    <table>...</table>
    <a href="/posts/new/">글 쓰기</a>
  </body>
</html>
```

Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING	ZAM	2020년 11월 3일 11:02 오전	보기
글 쓰기				

```
Elements Console Sources Network >> 2
<!DOCTYPE html>
<html>
  <head>
    <title>GSM</title>
    <link rel="stylesheet" href="/static/style.css" type="text/css">
    <link rel="stylesheet" href="/static/posts/style.css" type="text/css">
  </head>
  ... <body> == $0
    <h1>Posts</h1>
    <table>...</table>
    <a href="/posts/new/">글 쓰기</a>
  </body>
</html>
```

Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING ZAM		2020년 11월 3일 11:02 오전	보기
글 쓰기				

```
Elements Console Sources Network >> 2
<!DOCTYPE html>
<html>
  <head>
    <title>GSM</title>
    <link rel="stylesheet" href="/static/style.css" type="text/css">
    <link rel="stylesheet" href="/static/posts/style.css" type="text/css">
  </head>
  <body>
    <h1>Posts</h1>
    <table>...</table>
    <a href="/posts/new/">글 쓰기</a>
  </body>
</html>
```


https://getbootstrap.com/





Bootstrap · The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web.

getbootstrap.com

B

Home Documentation Examples Icons Themes Expo Blog

v4.5



Download


Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

Get started

Download

Currently v4.5.3





Getting started

Layout

Content

[Reboot](#)[Typography](#)[Code](#)[Images](#)[Tables](#)[Figures](#)

Components

[Utilities](#)[Extend](#)[Migration](#)[About](#)

Table head options

Similar to tables and dark tables, use the modifier classes `.thead-light` or `.thead-dark` to make `<thead>`s appear light or dark gray.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

[Copy](#)



Getting started

Layout

Content

Components

[Alerts](#)[Badge](#)[Breadcrumb](#)[Buttons](#)[Button group](#)[Card](#)[Carousel](#)[Collapse](#)[Dropdowns](#)

Forms

[Input group](#)[Jumbotron](#)[List group](#)[Media object](#)[Modal](#)[Navs](#)

```
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Form controls

Textual form controls—like `<input>`s, `<select>`s, and `<textarea>`s—are styled with the `.form-control` class. Included are styles for general appearance, focus state, sizing, and more.

Be sure to explore our [custom forms](#) to further style `<select>`s.

Email address

Example select

Example multiple select

Example textarea

Stacked to horizontal

Using a single set of `.col-sm-*` classes, you can create a basic grid system that starts out stacked and becomes horizontal at the small breakpoint (`sm`).

col-sm-8								col-sm-4			
col-sm				col-sm				col-sm			

```
<div class="container">
  <div class="row">
    <div class="col-sm-8">col-sm-8</div>
    <div class="col-sm-4">col-sm-4</div>
  </div>
  <div class="row">
    <div class="col-sm">col-sm</div>
    <div class="col-sm">col-sm</div>
    <div class="col-sm">col-sm</div>
  </div>
</div>
```

[Copy](#)

Quick start

Looking to quickly add Bootstrap to your project? Use jsDelivr, a free open source CDN. Using a package manager or need to download the source files? [Head to the downloads page](#).

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e
```

[Copy](#)

JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. We use [jQuery's slim build](#), but the full version is also supported.

Place **one of the following** `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.





base.html



```
1  {% load static %}
2
3  <!DOCTYPE html>
4  <html>
5      <head>
6          <title>GSM</title>
7          <link rel="stylesheet" href="{% static 'style.css' %}" type="text/css" />
8
9          {% block style %}
10             {% endblock %}
11      </head>
12      <body>
13          {% block content %}
14             {% endblock %}
15
16      </body>
17  </html>
```

CSS는 여기에

JS 파일들은 여기에

Quick start

Looking to quickly add Bootstrap to your project? Use jsDelivr, a free open source CDN. Using a package manager or need to download the source files? [Head to the downloads page](#).

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e
```

Copy

JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. We use [jQuery's slim build](#), but the full version is also supported.

Place **one of the following `<script>s`** near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

Bundle

Include everything you need in one script with our bundle. Our `bootstrap.bundle.js` and `bootstrap.bundle.min.js` include [Popper](#), but not [jQuery](#). For more information about what's included in Bootstrap, please see our [contents](#) section.

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVY"
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8Ah
```

[Copy](#)

Separate

If you decide to go with the separate scripts solution, Popper.js must come first, and then our JavaScript plugins.

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVY"
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66"
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzyY0tgp4pk
```

[Copy](#)

```
base.html x +
1 {% load static %}
2
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <title>GSM</title>
7
8     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/1I/4tG3dYVfMZTU46LJf9VUg6B7MC1KH0WgbU4d1Zs7K5DTp" crossorigin="anonymous">
9     {% block style %}
10    {% endblock %}
11  </head>
12  <body>
13    {% block content %}
14
15    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUj/42wK3sW+rZ" crossorigin="anonymous"></script>
16    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-c6780/8MmpoYQ6K7Uj2tLj3Ig9vFH0lY/QN/340i7a9/B9FXC0isZN6Zmn+AC/VqsZ60" crossorigin="anonymous"></script>
17    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-1jWZQcE86AB7iXBzFhg636s63eN7s63Eh437p8FQ6AH9tUg30Dy8P1Zb9joc" crossorigin="anonymous"></script>
18  </body>
19 </html>
```

Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING ZAM		2020년 11월 3일 11:02 오전	보기
글 쓰기				

Index.html(기존)



Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING ZAM		2020년 11월 3일 11:02 오전	보기
글 쓰기				

Index.html(Bootstrap 설치 후)

Posts

#작성자	본문	게시일	상세
1minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6CODING	ZAM	2020년 11월 3일 11:02 오전	보기

[글 쓰기](#)

[Getting started](#)[Layout](#)[Content](#)[Reboot](#)[Typography](#)[Code](#)[Images](#)[Tables](#)[Figures](#)[Components](#)[Utilities](#)[Extend](#)[Migration](#)[About](#)

Tables

[View on GitHub](#)

Documentation and examples for opt-in styling of tables (given their prevalent use in JavaScript plugins) with Bootstrap.



Limited time offer: Get 10 free Adobe Stock images.

ads via Carbon

Examples

Due to the widespread use of tables across third-party widgets like calendars and date pickers, we've designed our tables to be **opt-in**. Just add the base class `.table` to any `<table>`, then extend with custom styles or our various included modifier classes.

Using the most basic table markup, here's how `.table`-based tables look in Bootstrap. **All table styles are inherited in Bootstrap 4**, meaning any nested tables will be styled in the same manner as the parent.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
```

[Copy](#)

Table head options

Similar to tables and dark tables, use the modifier classes `.thead-light` or `.thead-dark` to make `<thead>`s appear light or dark gray.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table">
  <thead class="thead-dark">
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
```

Copy



index.html



```
1  {% extends "base.html" %}
2
3  {% load static %}
4
5  {% block style %}
6      <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
7  {% endblock %}
8
9  {% block content %}
10     <h1>Posts</h1>
11
12     {% if posts %}
13         <table>
14             <thead>
15                 <tr>
16                     <th>#</th>
17                     <th>작성 자</th>
18                     <th>본문</th>
19                     <th>게시일</th>
20                     <th>상세</th>
21                 </tr>
22             </thead>
23             <tbody>
24                 {% for post in posts %}
25                     <tr>
26                         <th>{{ post.id }}</th>
27                         <td>{{ post.author }}</td>
28                         <td>{{ post.body }}</td>
29                         <td>{{ post.created_at }}</td>
30                         <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
31                     </tr>
32                 {% endfor %}
33             </tbody>
34         </table>
35     {% else %}
36         <p>No Posts.</p>
37     {% endif %}
38
39     <a href="{% url 'posts:new' %}">글 쓰 기</a>
40 {% endblock %}
41
```



index.html



```
1 {% extends "base.html" %}
2
3 {% load static %}
4
5 {% block style %}
6     <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
7 {% endblock %}
8
9 {% block content %}
10     <h1>Posts</h1>
11
12     {% if posts %}
13         <table class="table">
14             <thead class="thead-dark">
15                 <tr>
16                     <th>#</th>
17                     <th>작성자</th>
18                     <th>본문</th>
19                     <th>게시일</th>
20                     <th>상세</th>
21                 </tr>
22             </thead>
23             <tbody>
24                 {% for post in posts %}
25                     <tr>
26                         <th>{{ post.id }}</th>
27                         <td>{{ post.author }}</td>
28                         <td>{{ post.body }}</td>
29                         <td>{{ post.created_at }}</td>
30                         <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
31                     </tr>
32                 {% endfor %}
33             </tbody>
34         </table>
35     {% else %}
36         <p>No Posts.</p>
37     {% endif %}
38
39     <a href="{% url 'posts:new' %}">글 쓰기</a>
40 {% endblock %}
41
```


Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING	ZAM	2020년 11월 3일 11:02 오전	보기

[글 쓰기](#)

Examples

Quickly get a project started with any of our examples ranging from using parts of the framework to custom components and layouts.

[Download examples](#)

[Download source code](#)



Custom components

Brand new components and templates to help folks quickly get started with Bootstrap and demonstrate best practices for adding onto the framework.



Album

Simple one-page template for photo galleries, portfolios, and more.



Pricing

Example pricing page built with Cards and featuring a custom header and footer.



Checkout

Custom checkout form showing our form components and their validation features.



Product

Lean product-focused marketing page with extensive grid and image work.



Cover

A one-page template for building simple and beautiful home pages.



Carousel

Customize the navbar and carousel, then add some new components.



Blog

Magazine-like blog template with header, navigation, featured content, and more.



Dashboard

Basic admin dashboard shell with fixed sidebar and navbar.

Posts

#	작성자	본문	게시일	상세
1	minyoung	나의 Django Project :D 수정하기를 시작합니다.	2020년 10월 12일 9:52 오전	보기
2	flower	안녕 나는 꽃이야	2020년 10월 12일 10:21 오전	보기
3	minil	Hello I am minil	2020년 10월 13일 10:48 오전	보기
6	CODING	ZAM	2020년 11월 3일 11:02 오전	보기

[글 쓰기](#)

Containers

Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**. Containers are used to contain, pad, and (sometimes) center the content within them. While containers can be nested, most layouts do not require a nested container.

Bootstrap comes with three different containers:

- `.container`, which sets a `max-width` at each responsive breakpoint
- `.container-fluid`, which is `width: 100%` at all breakpoints
- `.container-{breakpoint}`, which is `width: 100%` until the specified breakpoint

The table below illustrates how each container's `max-width` compares to the original `.container` and `.container-fluid` across each breakpoint.

See them in action and compare them in our [Grid example](#).

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
<code>.container</code>	100%	540px	720px	960px	1140px
<code>.container-sm</code>	100%	540px	720px	960px	1140px
<code>.container-md</code>	100%	100%	720px	960px	1140px
<code>.container-lg</code>	100%	100%	100%	960px	1140px
<code>.container-xl</code>	100%	100%	100%	100%	1140px
<code>.container-fluid</code>	100%	100%	100%	100%	100%



index.html



```
1 {% extends "base.html" %}
2
3 {% load static %}
4
5 {% block style %}
6     <link rel="stylesheet" href="{% static 'posts/style.css' %}" type="text/css" />
7 {% endblock %}
8
9 {% block content %}
10     <div class="container">
11         <h1>Posts</h1>
12
13         {% if posts %}
14             <table class="table">
15                 <thead class="thead-dark">
16                     <tr>
17                         <th>#</th>
18                         <th>작성자</th>
19                         <th>본문</th>
20                         <th>게시일</th>
21                         <th>상세</th>
22                     </tr>
23                 </thead>
24                 <tbody>
25                     {% for post in posts %}
26                         <tr>
27                             <th>{{ post.id }}</th>
28                             <td>{{ post.author }}</td>
29                             <td>{{ post.body }}</td>
30                             <td>{{ post.created_at }}</td>
31                             <td><a href="{% url 'posts:detail' post.id %}">보기</a></td>
32                         </tr>
33                     {% endfor %}
34                 </tbody>
35             </table>
36         {% else %}
37             <p>No Posts.</p>
38         {% endif %}
39
40         <a href="{% url 'posts:new' %}">글 쓰기</a>
41     </div>
42 {% endblock %}
```

← → ↻ 🔒 7499bd749db844dbb360c7bcc6b805e1.vfs.cloud9.us-east-1.amazon... ☆ 🖨 ⚙ 👤 민영 ⋮

← → ↻ 🔒 7499bd749db844dbb360c7bcc6b805e1.vfs.cloud9.us-east-1.amazon... ☆ 🖨 ⚙ 👤 민영 ⋮

← → ↻ 🔒 7499bd749db844dbb360c7bcc6b805e1.vfs.cloud9.us-east-1.amazon... ☆ 🖨 ⚙ 👤 민영 ⋮

Getting started

Layout

[Overview](#)[Grid](#)[Utilities for layout](#)[Content](#)[Components](#)[Utilities](#)[Extend](#)[Migration](#)[About](#)

Mix and match

Don't want your columns to simply stack in some grid tiers? Use a combination of different classes for each tier as needed. See the example below for a better idea of how it all works.

.col-md-8

.col-6 .col-md-4

.col-6 .col-md-4

.col-6 .col-md-4

.col-6 .col-md-4

.col-6

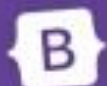
.col-6

```
<div class="container">
  <!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```

[Copy](#)



The Bootstrap Blog

News and announcements for all things **Bootstrap**, including new releases,
Bootstrap Themes, and **Bootstrap Icons**.

[All posts](#) [Archive](#) [Videos](#) [Subscribe](#)

Bootstrap 5 alpha!

16 Jun 2020



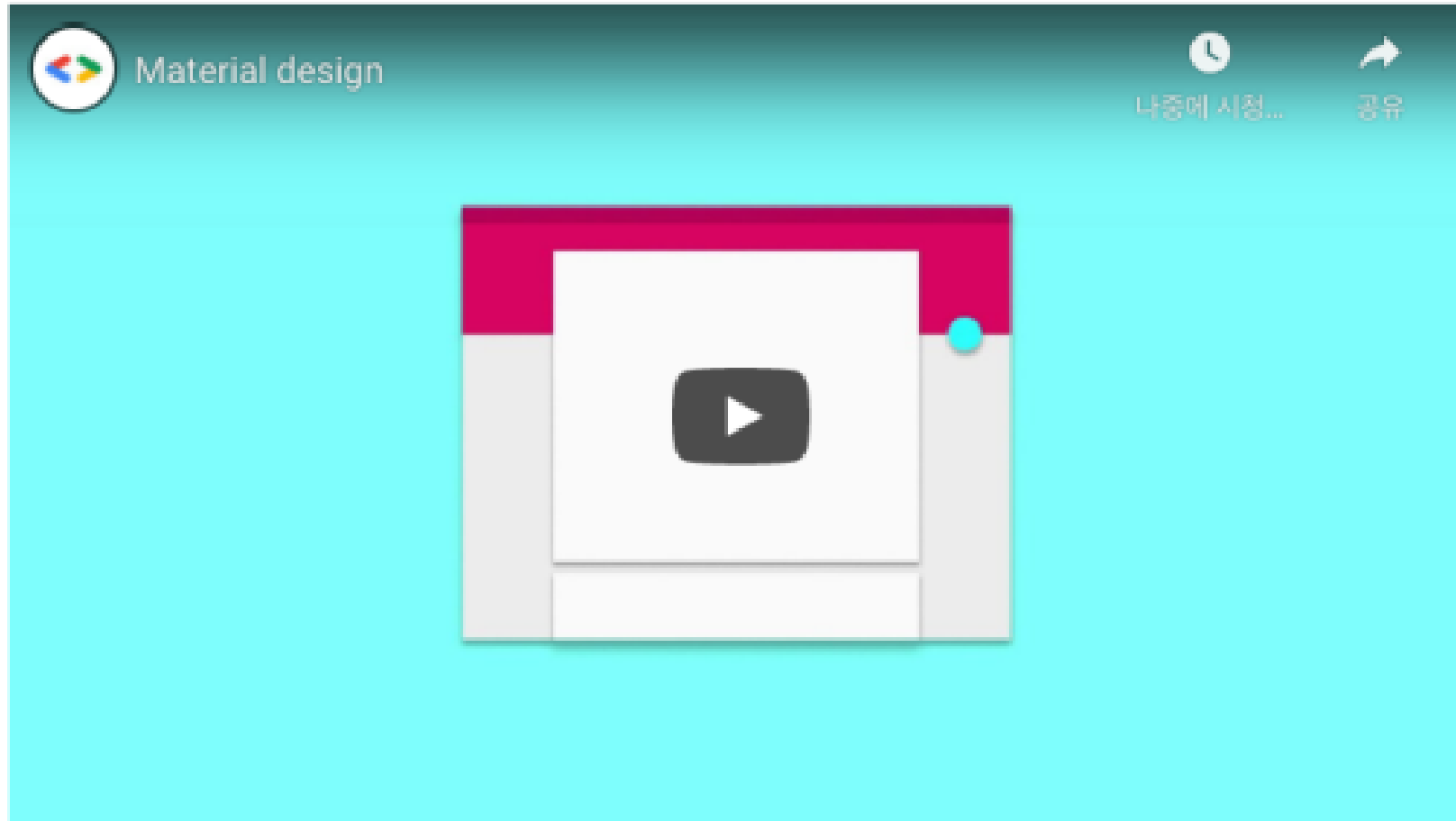
Bootstrap 5's very first alpha has arrived! We've been working hard for several months to refine the work we started in v4, and while we're feeling great about our progress, there's still even more to do.

We've been focused on making the migration from v4 to v5 more approachable, but we've also not been afraid to step away from what's outdated or no longer appropriate. As such, we're very happy to say that with v5, **Bootstrap no longer depends on jQuery** and **we've dropped support for Internet Explorer**. We're sharpening our focus on building tools that are more future-friendly, and while we're not fully there yet, the promise of CSS variables, faster JavaScript, fewer dependencies, and better APIs certainly feel right to us.

Before you jump to updating, please remember v5 is now in alpha—breaking changes will continue to occur until our first beta. We haven't finished adding or removing everything, so check for open issues and pull requests as you have questions or feedback.

Material Design

Created and designed by Google, Material Design is a design language that combines the classic principles of successful design along with innovation and technology. Google's goal is to develop a system of design that allows for a unified user experience across all their products on any platform.



Bootstrap Templates & Themes

Free Bootstrap themes that are ready to customize and publish. All of our themes are built with Bootstrap 4, MIT licensed, and updated regularly!



Premium Bootstrap themes,
templates, and UI kits by
Creative Tim.
Join our Carbon



SB Admin 2

A free Bootstrap admin theme

Free



SB UI Kit Pro

A premium Bootstrap 4 UI kit

Pro



SB Admin Pro

A premium Bootstrap admin theme

Pro



SB UI Kit Pro Angular

A premium Angular 9 UI kit

Pro



SB Admin Pro Angular

A premium Angular 9 admin template

Pro



Freelancer

A one page freelancer theme

Free



Creative

A one page creative theme

Free



Clean Blog - Angular

Blog built with Angular 9 and Node.js

Free



Agency

A one page agency theme

Free

END

**Update & Delete / Template Inheritance /
Bootstrap**

끝 !

더 배워야 할 것

:

USER 모델,

데이터베이스 관계 맺기(1:N,N:M,1:1)

Media, Git, API, Crawling, SQL연동,

도메인구매, 배포 등.. 내년.. 기회가되면,,