

파이썬 웹 개발부터 배포까지!

점프 투 장고



장고의 기본 요소 익히기!

INDEX

- 02-1 주소와 화면을 연결하는 URL과 뷰
- 02-2 데이터를 관리하는 모델
- 02-3 개발 편의를 제공하는 장고 Admin
- 02-4 질문 목록과 질문 상세 기능 구현하기
- 02-5 URL 더 똑똑하게 사용하기



장고의 기본 요소 익히기!

INDEX

02-6 답변 등록 기능 만들기

02-7 화면 예쁘게 꾸미기

02-8 부트스트랩으로 더 쉽게 화면 꾸미기

02-9 표준 HTML과 템플릿 상속 사용해 보기

02-10 질문 등록 기능 만들기



장고의 기본 요소 익히기!

이장의 목표

- ✓ urls.py 파일을 이용해 URL과 매핑되는 뷰 함수를 관리한다.
- ✓ 장고 ORM을 이용해 데이터베이스를 제어한다.
- ✓ 파이보 게시판에 질문 목록과 질문 상세 기능을 만든다.

02-7 화면 예쁘게 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-07

Do it!
실습

웹 페이지에 스타일시트 적용하기

01단계 설정 파일에 스타틱 디렉터리 위치 추가하기

```
파일 이름 C:/projects/mysite/config/settings.py

(... 생략 ...)
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    BASE_DIR / 'static',
]
```

STATIC_URL 항목은 맨 아래에 있음!

- ▶ BASE_DIR / 'static'은 C:/projects/mysite/static을 의미한다.

02단계 스타틱 디렉터리 만들고 스타일시트 작성하기

```
G:\ 명령 프롬프트
(mysite) C:/projects/mysite>mkdir static

파일 이름 C:/projects/mysite/static/style.css

textarea {
    width:100%;
}

input[type=submit] {
    margin-top:10px;
}
```

- ▶ 답변을 등록할 때 사용하는. textarea를 100%로 넓히고, <답변등록> 버튼 위에 margin을 10px 추가했다.

02-7 화면 예쁘게 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-07

Do it!
실습

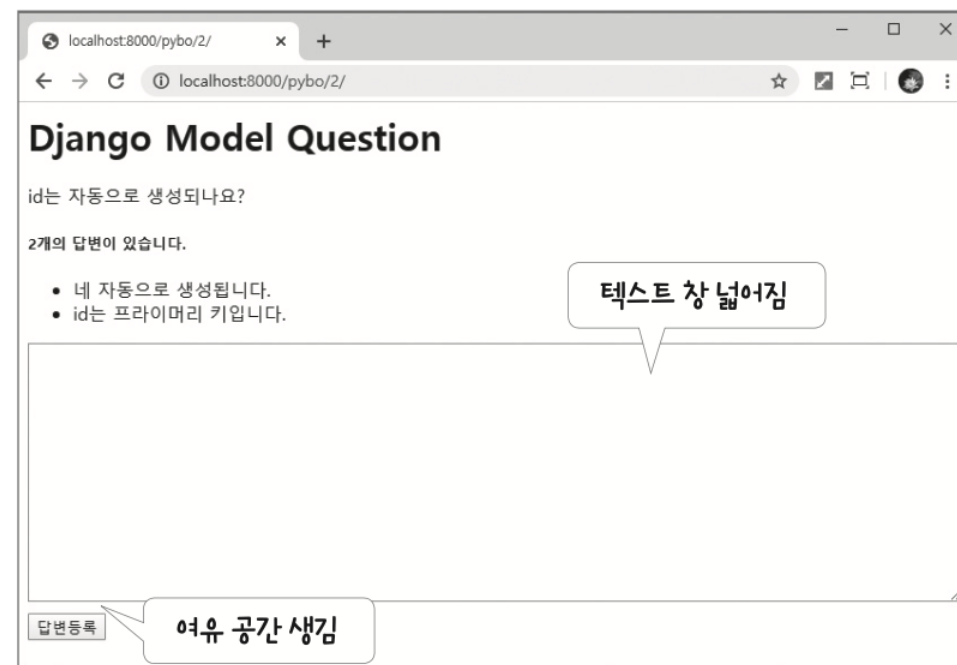
웹 페이지에 스타일시트 적용하기

03단계 질문 상세 템플릿에 스타일 적용하기

파일 이름 C:/projects/mysite/templates/pybo/question_detail.html

```
{% load static %}
<link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">
<h1>{{ question.subject }}</h1>
(... 생략 ...)
```

- ▶ 스태틱 파일을 사용하기 위해 템플릿 파일 맨 위에 {% load static %} 태그를 삽입하고, link 엘리먼트 href 속성에 {% static 'style.css' %}를 적자.
- ▶ 추가한 코드는 static 디렉터리의 style.css 파일을 연결한다는 의미다.



변경된 답변 등록 화면

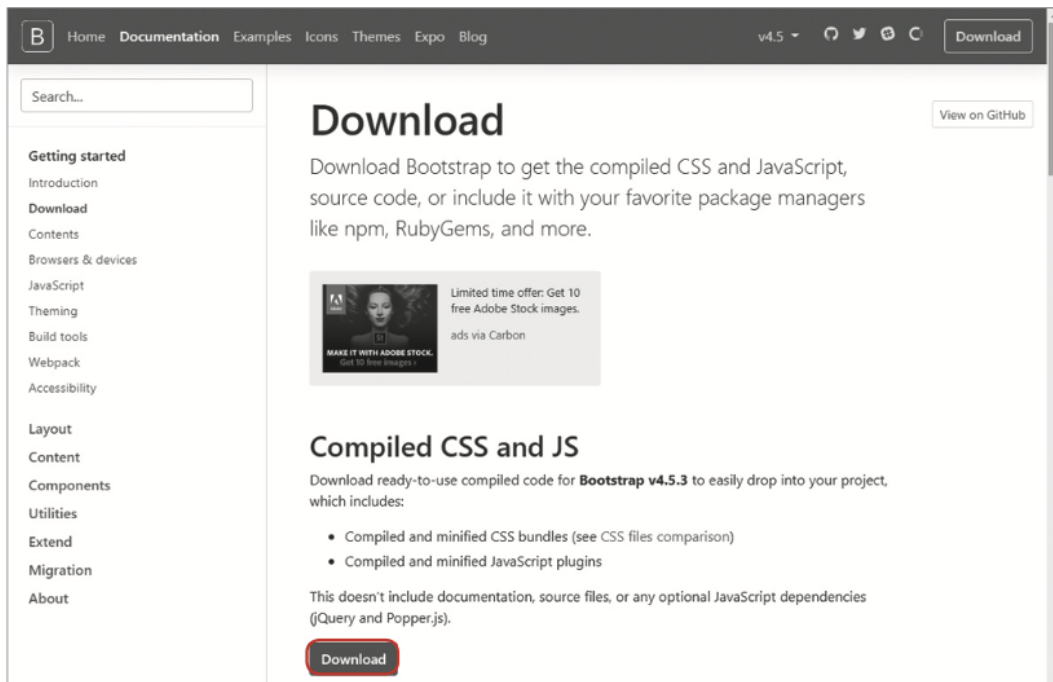
02-8 부트스트랩으로 더 쉽게 화면 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-08

Do it!
실습

파이보에 부트스트랩 적용하기

01단계 부트스트랩 설치하기



- ▶ 부트스트랩 4.5.3 버전 다운로드
: getbootstrap.com/docs/4.5/getting-started/download
- ▶ 내려받은 파일의 압축을 해제한 후 bootstrap.min.css 파일만 복사해서 mysite/static 디렉터리에 저장하자.

- 압축 파일에 있는 bootstrap.min.css 경로
`bootstrap-4.5.3-dist.zip/bootstrap-4.5.3-dist/css/bootstrap.min.css`
- 복사할 경로
`C:/projects/mysite/static/bootstrap.min.css`

02-8 부트스트랩으로 더 쉽게 화면 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-08

Do it!
실습

파이보에 부트스트랩 적용하기

02단계 질문 목록 템플릿에 부트스트랩 적용하기

파일 이름 C:/projects/mysite/templates/pybo/question_list.html

```
{% load static %}
<link rel="stylesheet" type="text/css" href="{% static 'bootstrap.min.css' %}">
{% if question_list %}
(... 생략 ...)
```

- ▶ pybo/question_list.html 파일에 부트스트랩을 적용하기 위해
{% load static %} 태그와 link 엘리먼트를 추가하자.

파일 이름 C:/projects/mysite/templates/pybo/question_list.html

```
{% load static %}
<link rel="stylesheet" type="text/css" href="{% static 'bootstrap.min.css' %}">
<div class="container my-3">
  <table class="table">
    <thead>
      <tr class="thead-dark">
        <th>번호</th>
        <th>제목</th>
        <th>작성일시</th>
      </tr>
    </thead>
    <tbody>
      {% if question_list %}
      {% for question in question_list %}
      <tr>
        <td>{{ forloop.counter }}</td>
```


02-8 부트스트랩으로 더 쉽게 화면 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-08

Do it!
실습

파이보에 부트스트랩 적용하기

02단계 질문 목록 템플릿에 부트스트랩 적용하기

```
<td>
  <a href="{% url 'pybo:detail' question.id %}">
    {{ question.subject }}
  </a>
</td>
<td>{{ question.create_date }}</td>
</tr>
{% endfor %}
{% else %}
<tr>
  <td colspan="3">질문이 없습니다.</td>
</tr>
{% endif %}
</tbody>
</table>
</div>
```

- ▶ 여기서 사용된 container, my-3, thead-dark 등이 바로 부트스트랩이 제공하는 클래스이다.
- ▶ {{ forloop.counter }}는 {% for... %}에서 반복 시 자동으로 매겨지는 순서값을 의미한다.



부트스트랩이 적용된 질문 목록 조회 화면

02-8 부트스트랩으로 더 쉽게 화면 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-08

Do it!
실습

파이보에 부트스트랩 적용하기

03단계 질문 상세 템플릿에 부트스트랩 사용하기

파일 이름 C:/projects/mysite/templates/pybo/question_detail.html

```
{% load static %}
<link rel="stylesheet" type="text/css" href="{% static 'bootstrap.min.css' %}">
<div class="container my-3">
  <h2 class="border-bottom py-2">{{ question.subject }}</h2>
  <div class="card my-3">
    <div class="card-body">
      <div class="card-text" style="white-space: pre-line;">
        {{ question.content }}
      </div>
      <div class="d-flex justify-content-end">
        <div class="badge badge-light p-2">
          {{ question.create_date }}
        </div>
      </div>
    </div>
  </div>
</div>
```

```
<h5 class="border-bottom my-3 py-2">
  {{ question.answer_set.count }}개의 답변이 있습니다.
</h5>
{% for answer in question.answer_set.all %}
<div class="card my-3">
  <div class="card-body">
    <div class="card-text" style="white-space: pre-line;">
      {{ answer.content }}
    </div>
    <div class="d-flex justify-content-end">
      <div class="badge badge-light p-2">
        {{ answer.create_date }}
      </div>
    </div>
  </div>
</div>
{% endfor %}
```

02-8 부트스트랩으로 더 쉽게 화면 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-08

Do it!
실습

파이보에 부트스트랩 적용하기

03단계 질문 상세 템플릿에 부트스트랩 사용하기

```
<form action="{% url 'pybo:answer_create' question.id %}"  
      method="post" class="my-3">  
  {% csrf_token %}  
  <div class="form-group">  
    <textarea name="content" id="content"  
              class="form-control" rows="10"></textarea>  
  </div>  
  <input type="submit" value="답변 등록" class="btn btn-primary">  
</form>  
</div>
```

- ▶ 질문, 답변은 하나의 문치에 해당되므로 부트스트랩의 card 컴포넌트를 사용했다.
- ▶ 질문 내용과 답변 내용은 style속성으로 white-space: pre-line을 적용하여 텍스트의 줄바꿈을 정상적으로 보이게 만들었다.
- ▶ my-3 : 상하 마진값 3
py-2 : 상하 패딩값 2
p-2 : 상하좌우 패딩값 2
d-flex justify-content-end : 컴포넌트 오른쪽 정렬

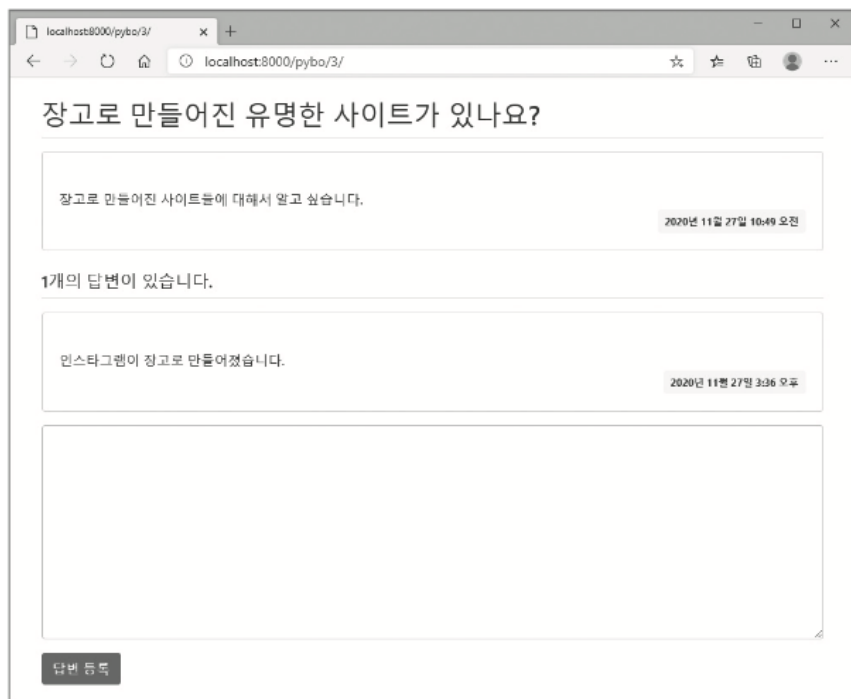
02-8 부트스트랩으로 더 쉽게 화면 꾸미기

• 완성 소스 github.com/pahkey/djangobook/tree/2-08

Do it!
실습

파이보에 부트스트랩 적용하기

04단계 질문 상세 화면 확인하기



부트스트랩이 적용된 질문 상세 화면

02-9 표준 HTML과 템플릿 상속 사용해 보기

• 완성 소스 github.com/pahkey/djangobook/tree/2-09

표준 HTML 구조는 어떻게 생겼을까?

- ▶ 표준 HTML 문서의 구조는 html, head, body 엘리먼트가 있어야 하며, CSS 파일은 head 엘리먼트 안에 있어야 한다.
- ▶ head 엘리먼트 안에는 meta, title 엘리먼트 등이 포함되어야 한다.

```
표준 HTML 구조 예

{% load static %}
<!doctype html>
<html lang="ko">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" type="text/css" href="{% static 'bootstrap.min.css' %}">
  <!-- pybo CSS -->
  <link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">
  <title>Hello, pybo!</title>
</head>
<body>
  (... 생략 ...)
</body>
</html>
```

입력할 코드 아님!

02-9 표준 HTML과 템플릿 상속 사용해 보기

• 완성 소스 github.com/pahkey/djangobook/tree/2-09

Do it!
실습

템플릿을 표준 HTML 구조로 바꾸기

01단계 템플릿 파일의 기본 틀 작성하기

```
파일 이름 C:/projects/mysite/templates/base.html

{% load static %}
<!doctype html>
<html lang="ko">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" type="text/css" href="{% static 'bootstrap.min.css' %}">
  <!-- pybo CSS -->
  <link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">
  <title>Hello, pybo!</title>
</head>
<body>
<!-- 기본 템플릿 안에 삽입될 내용 Start -->
{% block content %}
{% endblock %}
```

```
<!-- 기본 템플릿 안에 삽입될 내용 End -->
</body>
</html>
```

- ▶ 장고는 템플릿 상속(extends) 기능을 제공한다.
- ▶ {% block content %}와 {% endblock %} 템플릿 태그는 이후 base.html 템플릿 파일을 상속한 파일에서 구현해야 하는 영역이 된다.

02-9 표준 HTML과 템플릿 상속 사용해 보기

• 완성 소스 github.com/pahkey/djangobook/tree/2-09

Do it!
실습

템플릿을 표준 HTML 구조로 바꾸기

02단계 질문 목록 템플릿 수정하기

```
파일 이름 C:/projects/mysite/templates/pybo/question_list.html

{% extends 'base.html' %}
{% block content %}
<div class="container my-3">
  <table class="table">
    (... 생략 ...)
  </table>
</div>
{% endblock %}
```

기존 코드 삭제 후 추가

새로 추가

- ▶ question_list.html은 base.html을 상속받았으므로 표준 HTML 구조를 갖추게 되었다.

03단계 질문 상세 템플릿 수정하기

```
파일 이름 C:/projects/mysite/templates/pybo/question_detail.html

{% extends 'base.html' %}
{% block content %}
<div class="container my-3">
  <h2 class="border-bottom py-2">{{ question.subject }}</h2>
  (... 생략 ...)
</form>
</div>
{% endblock %}
```

기존 코드 삭제 후 추가

새로 추가

- ▶ {% extends 'base.html' %} 템플릿 태그를 맨 위에 추가하고 기존 내용 위 아래로 {% block content %}와 {% endblock %}를 작성했다.

02-9 표준 HTML과 템플릿 상속 사용해 보기

• 완성 소스 github.com/pahkey/djangobook/tree/2-09

Do it!
실습

템플릿을 표준 HTML 구조로 바꾸기

04단계 기존 스타일 파일 내용 비우기

파일 이름 C:/projects/mysite/static/style.css

```
/* 내용을 전부 삭제하자. */
```

- ▶ 이 파일은 이후 부트스트랩으로 표현할 수 없는 스타일을 위해 사용할 것이므로 파일 자체를 삭제하지는 말고 내용만 삭제하자.

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

01단계 질문 등록 버튼 만들기

```
파일 이름 C:/projects/mysite/templates/pybo/question_list.html

(... 생략 ...)
</table>
<a href="{% url 'pybo:question_create' %}" class="btn btn-primary">
    질문 등록하기
</a>
</div>
```

- ▶ a 엘리먼트에 href 속성으로 질문 등록 URL을 {% url 'pybo:question_create' %}처럼 추가하고 부트스트랩 클래스 "btn btn-primary"를 지정했다.

02단계 URL 매핑 추가를 위해 pybo/urls.py 수정하기

```
파일 이름 C:/projects/mysite/pybo/urls.py

(... 생략 ...)
urlpatterns = [
    (... 생략 ...)
    path('question/create/', views.question_create, name='question_create'),
]
```

pybo 디렉터리

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

03단계 pybo/views.py 수정하기

```
파일 이름 C:/projects/mysite/pybo/views.py

from .forms import QuestionForm
(... 생략 ...)
def question_create(request):
    """
    pybo 질문 등록
    """
    form = QuestionForm()
    return render(request, 'pybo/question_form.html', {'form': form})
```

- ▶ QuestionForm 클래스는 질문을 등록하기 위해 사용하는 장고의 폼이다.
- ▶ render 함수에 전달한 {'form' : form}은 템플릿에서 폼 엘리먼트를 생성할 때 사용한다.

04단계 pybo/forms.py에 장고 폼 작성하기

```
파일 이름 C:/projects/mysite/pybo/forms.py

from django import forms
from pybo.models import Question
class QuestionForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['subject', 'content']
```

- ▶ forms.ModelForm을 상속받아 모델 폼을 만들었다.
- ▶ 장고 모델 폼은 내부 클래스로 Meta 클래스를 반드시 가져야 하며, Meta 클래스에는 모델 폼이 사용할 모델과 모델의 필드들을 적어야 한다.

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

05단계 pybo/question_form.html 만들어 장고 폼 사용하기

```
파일 이름 C:/projects/mysite/templates/pybo/question_form.html

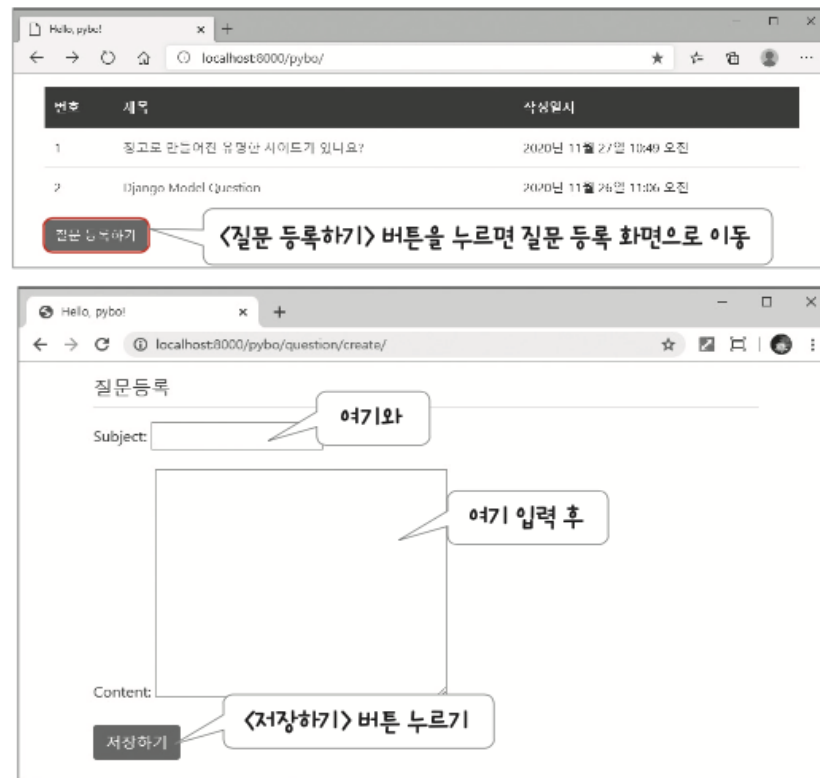
{% extends 'base.html' %}

{% block content %}
<div class="container">
  <h5 class="my-3 border-bottom pb-2">질문 등록</h5>
  <form method="post" class="post-form my-3">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit" class="btn btn-primary">저장하기</button>
  </form>
</div>
{% endblock %}
```

디렉터리 확인!

- ▶ {{ form.as_p }}는 모델 폼과 연결된 입력 항목 subject, content에 값을 입력할 수 있는 HTML 코드를 자동으로 만들어 준다.

06단계 질문 등록 화면 확인하기



02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

07단계 입력 데이터 저장하기

```
파일 이름 C:/projects/mysite/pybo/views.py

def question_create(request):
    """
    pybo 질문 등록
    """
    if request.method == 'POST':
        form = QuestionForm(request.POST)
        if form.is_valid():
            question = form.save(commit=False)
            question.create_date = timezone.now()
            question.save()
            return redirect('pybo:index')
    else:
        form = QuestionForm()
        context = {'form': form}
        return render(request, 'pybo/question_form.html', context)
```

request.method가 'GET'인 경우 호출!

- ▶ 동일한 URL 요청을 POST, GET 요청 방식에 따라 다르게 처리한다.
- ▶ QuestionForm 객체도 GET방식과 POST방식일 경우 다르게 생성한다.
- ▶ `question = form.save(commit=False)`는 form으로 Question 모델 데이터를 저장하기 위한 코드이다.
- ▶ 임시 저장을 사용하는 이유는 폼으로 질문 데이터를 저장할 경우 Question모델의 `create_date`에 값이 설정되지 않아 오류가 발생하기 때문이다.

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

08단계 질문 등록 기능 확인하기

The left screenshot shows a browser window at localhost:8000/pybo/. It contains a table with two rows of questions. Below the table is a button labeled '질문 등록하기'. A red arrow points from this button to the '저장하기' button in the right screenshot.

번호	제목	작성일시
1	장고로 만들어진 유명한 사이트가 있나요?	2020년 11월 27일 10:49 오전
2	Django Model Question	2020년 11월 26일 11:06 오전

The right screenshot shows the '질문 등록' form at localhost:8000/pybo/question/crea. It has a 'Subject' field with '산수질문' and a 'Content' text area with '1+1은 무엇인가요?'. A '저장하기' button is at the bottom right.

<질문 등록하기> 버튼 누르기

값 입력 후 <저장하기> 버튼 누르기

The screenshot shows the same question list page as before, but now with three rows. The first row, '산수질문', is highlighted with a red box. A callout bubble points to it with the text '새 게시물 확인'.

번호	제목	작성일시
1	산수질문	2020년 11월 27일 4:48 오후
2	장고로 만들어진 유명한 사이트가 있나요?	2020년 11월 27일 10:49 오전
3	Django Model Question	2020년 11월 26일 11:06 오전

질문 등록 완료

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

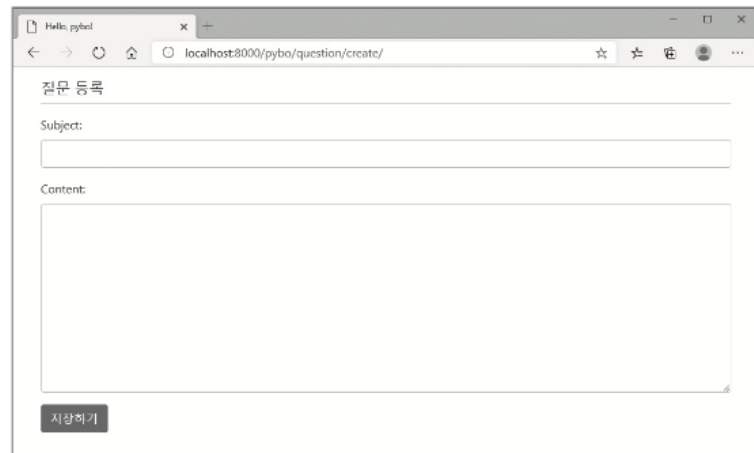
09단계 폼에 부트스트랩 적용하기

```
파일 이름 C:/projects/mysite/pybo/forms.py

from django import forms
from pybo.models import Question

class QuestionForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['subject', 'content']
        widgets = {
            'subject': forms.TextInput(attrs={'class': 'form-control'}),
            'content': forms.Textarea(attrs={'class': 'form-control', 'rows': 10}),
        }
```

- ▶ `{{ form.as_p }}` 태그는 form 엘리먼트와 입력 항목을 자동으로 생성해주므로 편리하기는 하지만 부트스트랩을 적용할 수 없다는 단점이 있다.



부트스트랩이 적용된 질문 등록 화면

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

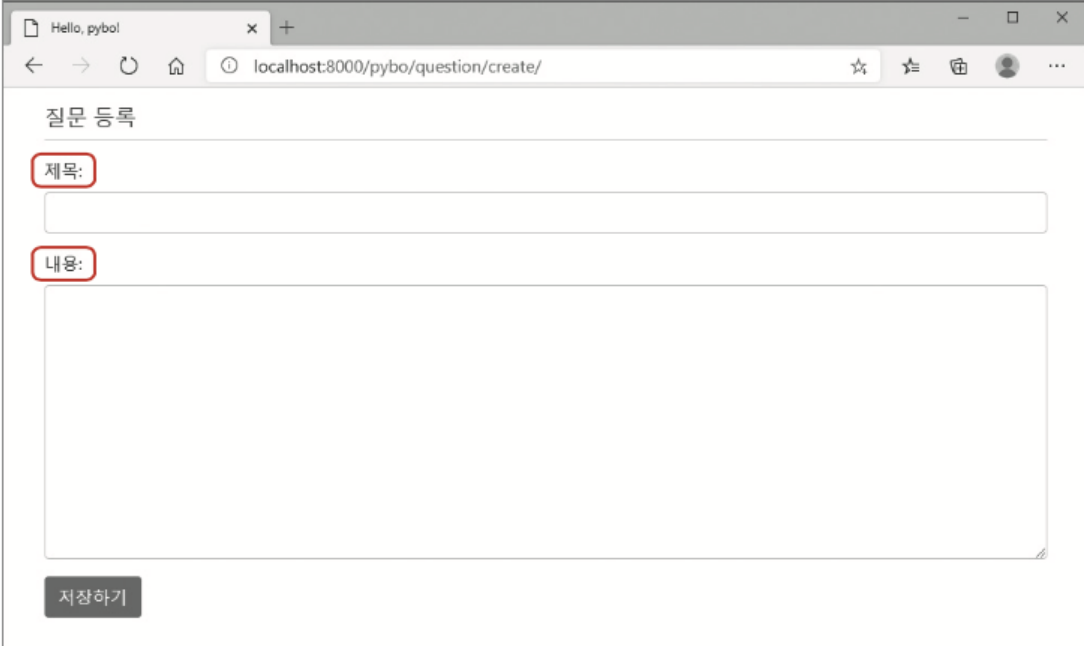
Do it!
실습

질문 등록 기능 만들기

10단계 label 속성 수정하여 Subject, Content 한글로 변경하기

```
from django import forms
from pybo.models import Question

class QuestionForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['subject', 'content']
        widgets = {
            'subject': forms.TextInput(attrs={'class': 'form-control'}),
            'content': forms.Textarea(attrs={'class': 'form-control', 'rows': 10}),
        }
        labels = {
            'subject': '제목',
            'content': '내용',
        }
```



한글로 변경된 질문 등록 화면

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

11단계 수작업으로 폼 작성하기

```
파일 이름 C:/projects/mysite/pybo/forms.py

from django import forms
from pybo.models import Question

class QuestionForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['subject', 'content']
        labels = {
            'subject': '제목',
            'content': '내용',
        }
# widget 항목 삭제
```

```
파일 이름 C:/projects/mysite/templates/pybo/question_form.html

{% extends 'base.html' %}

{% block content %}
<div class="container">
  <h5 class="my-3 border-bottom pb-2">질문 등록</h5>
  <form method="post" class="post-form my-3">
    {% csrf_token %}
    <!-- 오류 표시 Start -->
    {% if form.errors %}
      <div class="alert alert-danger" role="alert">
        {% for field in form %}

```

- ▶ forms.py 파일의 widget 항목을 제거하자.

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

```
{% if field.errors %}
<strong>{{ field.label }}</strong>
{{ field.errors }}
{% endif %}
{% endfor %}
</div>
{% endif %}
<!-- 오류 표시 End -->
<div class="form-group">
  <label for="subject">제목</label>
  <input type="text" class="form-control" name="subject" id="subject"
    value="{{ form.subject.value|default_if_none:'' }}">
</div>
<div class="form-group">
  <label for="content">내용</label>
  <textarea class="form-control" name="content" id="content"
    rows="10">{{ form.content.value|default_if_none:'' }}</textarea>
</div>
```

```
<button type="submit" class="btn btn-primary">저장하기</button>
</form>
</div>
{% endblock %}
```

- ▶ `{{ form.as_p }}`에 의해 자동 생성되는 HTML 대신 제목과 내용을 위한 HTML을 직접 작성했다.
- ▶ `question_create` 함수에서 `form.is_valid()`가 실패했을 때 오류를 표시하기 위해 오류 표시 영역도 추가했다.

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

질문 등록 기능 만들기

질문 등록

내용

- 필수 항목입니다.

제목

테스트

내용

필수 항목을 입력하지 않으면 나타나는 오류 메시지

- ▶ 필수 항목인 내용을 입력하지 않았으니 '내용을 입력하라'는 오류 메시지를 볼 수 있다.
- ▶ 오류가 발생해도 이미 입력한 제목에 해당되는 값은 value 설정으로 인해 그대로 유지되는 것도 확인할 수 있다.

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

답변 등록 기능에 장고 폼 적용하기

01단계 AnswerForm 클래스 추가하고 answer_create 함수 수정하기

```
파일 이름 C:/projects/mysite/pybo/forms.py

from django import forms
from pybo.models import Question, Answer
(... 생략 ...)
class AnswerForm(forms.ModelForm):
    class Meta:
        model = Answer
        fields = ['content']
        labels = {
            'content': '답변내용',
        }
```

- ▶ 질문 등록 기능에 장고 폼을 적용한 것처럼 답변 등록 기능에도 장고 폼을 적용하자.
- ▶ 답변을 등록할 때 사용할 AnswerForm클래스를 pybo/forms.py 파일에 다음과 같이 작성하자.

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

Do it!
실습

답변 등록 기능에 장고 폼 적용하기

파일 이름 C:/projects/mysite/pybo/views.py

```
(... 생략 ...)
from .forms import QuestionForm, AnswerForm
(... 생략 ...)

def answer_create(request, question_id):
    """
    pybo 답변 등록
    """
    question = get_object_or_404(Question, pk=question_id)

    if request.method == "POST":
        form = AnswerForm(request.POST)
        if form.is_valid():
            answer = form.save(commit=False)
            answer.create_date = timezone.now()
```

```
        answer.question = question
        answer.save()
        return redirect('pybo:detail', question_id=question.id)
    else:
        form = AnswerForm()
        context = {'question': question, 'form': form}
        return render(request, 'pybo/question_detail.html', context)
```

02-10 질문 등록 기능 만들기

• 완성 소스 github.com/pahkey/djangobook/tree/2-10

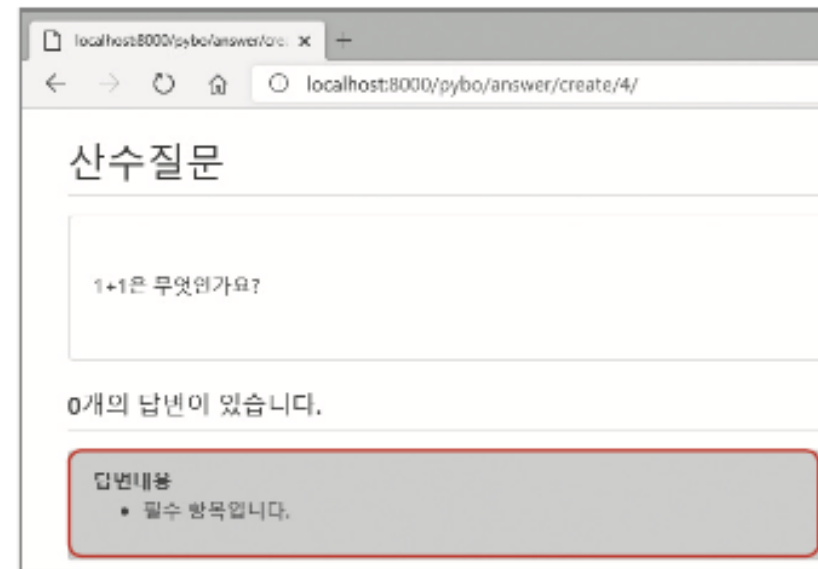
Do it!
실습

답변 등록 기능에 장고 폼 적용하기

02단계 질문 상세 템플릿에 오류 표시 영역 추가하기

```
파일 이름 C:/projects/mysite/templates/pybo/question_detail.html

(... 생략 ...)
<form action="{% url 'pybo:answer_create' question.id %}" method="post" class="my-3">
  {% csrf_token %}
  {% if form.errors %}
    <div class="alert alert-danger" role="alert">
      {% for field in form %}
        {% if field.errors %}
          <strong>{{ field.label }}</strong>
          {{ field.errors }}
        {% endif %}
      {% endfor %}
    </div>
  {% endif %}
</form>
(... 생략 ...)
```



답변 내용이 없으면 오류 메시지 나타남

Thank You