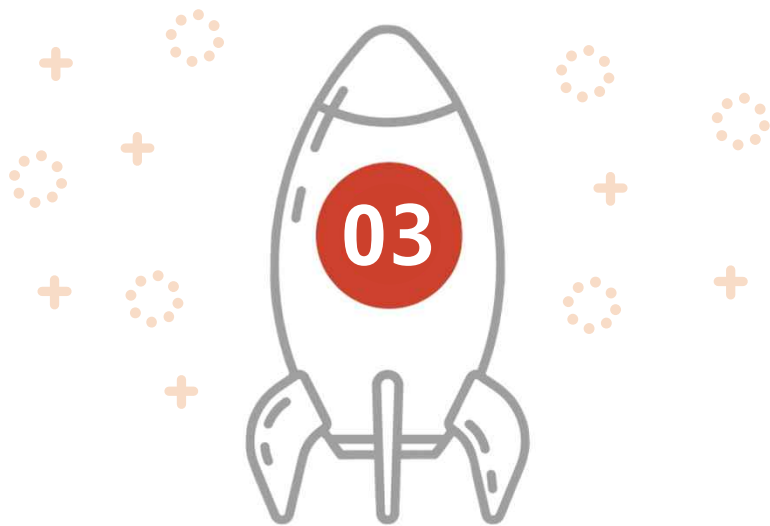


7주차 강의자료

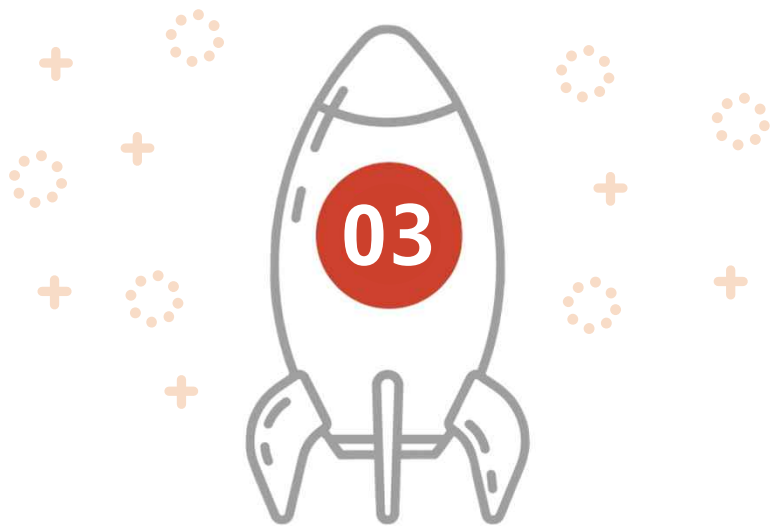
웹서버컴퓨팅



파이보 서비스 개발!

INDEX

- 03-1 내비게이션 기능 추가하기
- 03-2 게시판 페이징 기능 추가하기
- 03-3 템플릿 필터 직접 만들어 보기
- 03-4 질문에 달린 답변 개수 표시하기
- 03-5 로그인, 로그아웃 구현하기
- 03-6 회원가입 구현하기



파이보 서비스 개발!

INDEX

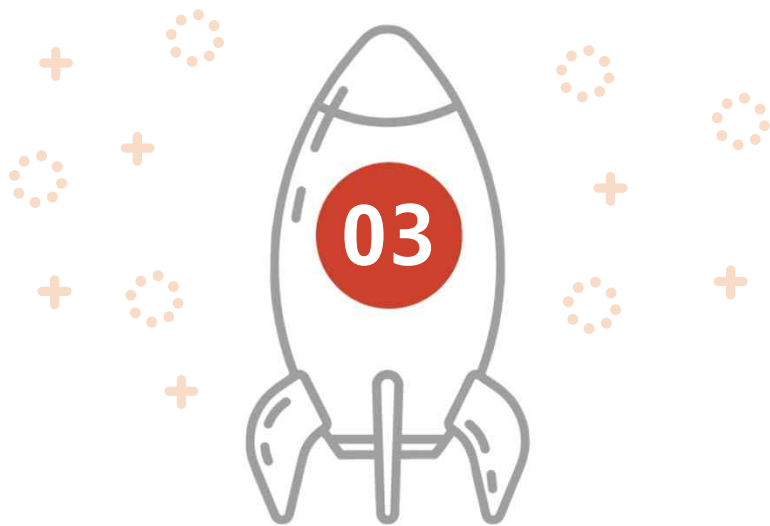
- 03-7 모델에 글쓰기 추가하기
- 03-8 글쓰기 표시하기
- 03-9 게시물 수정 & 삭제 기능 추가하기
- 03-10 댓글 기능 추가하기
- 03-11 views.py 파일 분리하기



파이보 서비스 개발!

INDEX

- 03-12 추천 기능 추가하기
- 03-13 스크롤 초기화 문제점 해결하기
- 03-14 마크다운 기능 적용하기
- 03-15 검색, 정렬 기능 추가하기
- 03-16 도전! 저자 추천 파이보 추가 기능



파이보 서비스 개발!

이장의 목표

- ✓ 파이보를 상용 게시판 수준으로 개발한다.
- ✓ 부트스트랩을 적용하여 서비스를 더 아름답게 만든다.
- ✓ 게시물 등록, 삭제, 수정부터 로그인, 로그아웃, 페이지, 검색까지 게시판을 완벽하게 만든다.

03-8 글쓴이 표시하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-08

Do it!
실습

질문 목록 화면에 글쓴이 표시하기

01단계 질문 목록 템플릿 수정하기

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

```
(... 생략 ...)  
<tr class="text-center thead-dark">  
  <th>번호</th>  
  <th style="width:50%">제목</th>  
  <th>글쓴이</th>  
  <th>작성일시</th>  
</tr>  
(... 생략 ...)
```

- ▶ 글쓴이를 표시하기 위해 테이블 헤더에 글쓴이 항목을 추가하자.

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

```
(... 생략 ...)  
{% for question in question_list %}  
<tr class="text-center">  
  <td>  
(... 생략 ...)  
</td>  
  <td class="text-left">  
    <a href="{% url 'pybo:detail' question.id %}">{{ question.subject }}</a>  
    {% if question.answer_set.count > 0 %}  
    <span class="text-danger small ml-2">{{ question.answer_set.count }}</span>  
    {% endif %}  
  </td>  
  <td>{{ question.author.username }}</td> <!-- 글쓴이 추가 -->  
  <td>{{ question.create_date }}</td>  
</tr>  
{% endfor %}  
(... 생략 ...)
```

03-8 글쓴이 표시하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-08

Do it!
실습

질문 상세 조회 화면에 글쓴이 표시하기

01단계 질문 상세 템플릿 수정하기

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

```
(... 생략 ...)
<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 text-left">
      <div class="mb-2">{{ question.author.username }}</div>
      <div>{{ question.create_date }}</div>
    </div>
  </div>
</div>
(... 생략 ...)
```

02단계 답변에 글쓴이 표시 기능 추가하기

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

```
(... 생략 ...)
<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 text-left">
      <div class="mb-2">{{ answer.author.username }}</div>
      <div>{{ answer.create_date }}</div>
    </div>
  </div>
</div>
(... 생략 ...)
```

- ▶ 글쓴이와 작성일시가 함께 보이도록 수정했다.

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

모델 수정하기

01단계 Question, Answer 모델에 modify_date 필드 추가하기

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

(... 생략 ...)

class Question(models.Model):
    (... 생략 ...)
    modify_date = models.DateTimeField(null=True, blank=True)
    (... 생략 ...)

class Answer(models.Model):
    (... 생략 ...)
    modify_date = models.DateTimeField(null=True, blank=True)
```

- ▶ null=True, blank=True는 어떤 조건으로든 값을 비워둘 수 있음을 의미한다.

02단계 makemigrations, migrate 명령 수행하기

```
C:\> 명령 프롬프트

(mysite) C:\projects\mysite>python manage.py makemigrations
Migrations for 'pybo':
pybo\migrations\0004_auto_20200331_0945.py
- Add field modify_date to answer
- Add field modify_date to question

(mysite) C:\projects\mysite>python manage.py migrate
Operations to perform:
Apply all migrations: admin, auth, contenttypes, pybo, sessions
Running migrations:
Applying pybo.0004_auto_20200331_0945... OK
```


03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

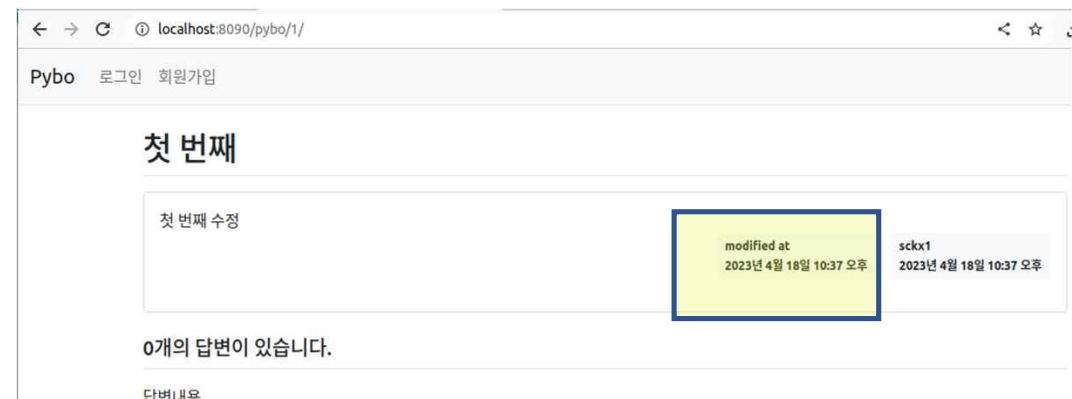
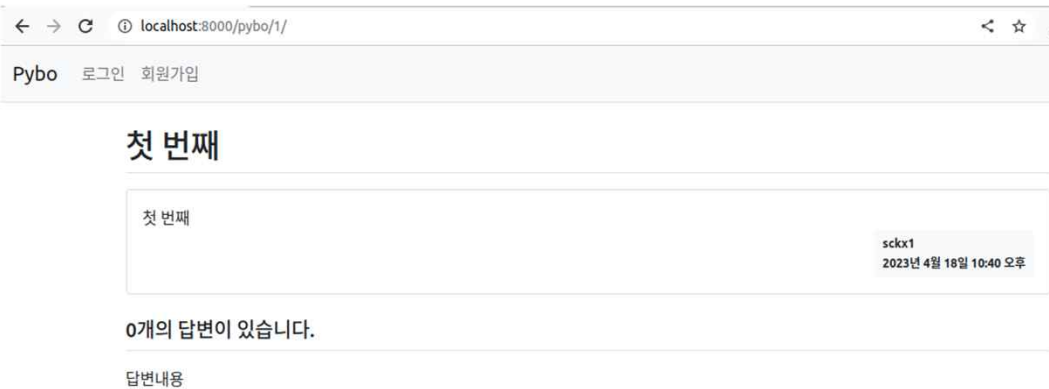
Do it!
실습

모델 수정하기

01단계 03-8의 pybo 앱에서 질문은 수정할 수 없다
<https://github.com/pahkey/jump2django/tree/3-08>



02단계 03-9의 pybo 앱에서 질문은 수정할 수 있다
<https://github.com/pahkey/jump2django/tree/3-09>



03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

모델 수정하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 modify_date 추가

- ▶ <https://github.com/pahkey/jump2django/tree/3-08>에서 modify_date를 pybo/models.py에서 추가한다.

```
3 |
4 |
5 | class Question(models.Model):
6 |     author = models.ForeignKey(User, on_delete=models.CASCADE)
7 |     subject = models.CharField(max_length=200)
8 |     content = models.TextField()
9 |     create_date = models.DateTimeField()
10 |     modify_date = models.DateTimeField(null=True, blank=True)
11 |
```

- ▶ makemigrations을 진행한다.

```
^C^(mysite) sckx1@sckx1-ThinkPad-X1-Carbon-Gen-8:~/projects/test/jump2django-3-08$ python manage.py makemigrations
Migrations for 'pybo':
  pybo/migrations/0004_answer_modify_date_question_modify_date.py
    - Add field modify_date to answer
    - Add field modify_date to question
```

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

모델 수정하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 0004 answer_modify_date_question_modify_date migrations 생성

0002_question_author.py
0003_answer_author.py
0004_answer_modify_date_question_modify_date.py
__init__.py
emplatetags
__init__.py
dmin.py
pps.py
orms.py
models.py
ests.py
rls.py
iews.py
ic
plates
ommon
ybo
question_detail.html
question_form.html
question_list.html
ase.html
orm_errors.html

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```
class Migration(migrations.Migration):

    dependencies = [
        ('pybo', '0003_answer_author'),
    ]

    operations = [
        migrations.AddField(
            model_name='answer',
            name='modify_date',
            field=models.DateTimeField(blank=True, null=True),
        ),
        migrations.AddField(
            model_name='question',
            name='modify_date',
            field=models.DateTimeField(blank=True, null=True),
        ),
    ]
```

▶ migrate를 진행한다.

```
(mysite) sckx1@sckx1-ThinkPad-X1-Carbon-Gen-8:~/projects/test/jump2django-3-08$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, pybo, sessions
Running migrations:
  Applying pybo.0004_answer_modify_date_question_modify_date... OK
```

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

모델 수정하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 질문 수정 버튼 생성

- ▶ `pybo/question_detail.html`에 질문 수정 버튼을 만들어야 한다.

두 번째

두 번째

수정

삭제

- ▶ `card-body`안에 하나의 배지와 하나의 수정버튼이 만들어진다.
- ▶ 배지안에 `question.author.username`을 표시하는 객체와 `create_date`를 표시하는 객체, 두 객체가 배지 안에 위치한다.

```
<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 bg-light text-dark p-2 text-start">
      <div class="mb-2">{{ question.author.username }}</div>
      <div>{{ question.create_date }}</div>
    </div>
  </div>

  <div class="my-3">
    {% if request.user == question.author %}
    <a href=" {% url 'pybo:question_modify' question.id %}"
      class="btn btn-sm btn-outline-secondary">수정</a>
    {% endif %}
  </div>
</div>
```

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

모델 수정하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 질문 수정 url 매핑

- ▶ 수정 버튼에 대한 이벤트를 생성할 때, url 을 사용한다.
- ▶ url 별칭은 question_modify이다.
- ▶ 별칭은 pybo/urls.py에서 매핑하여 준다

```
urlpatterns = [  
    path('', views.index, name='index'),  
    path('<int:question_id>/', views.detail, name='detail'),  
    path('answer/create/<int:question_id>/', views.answer_create, name='answer_create'),  
    path('question/create/', views.question_create, name='question_create'),  
    path('question/modify/<int:question_id>/', views.question_modify,  
    name='question_modify'),  
]
```

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

모델 수정하기

01단계 질문 수정은 기존의 질문 등록 폼 사용

- ▶ 질문 등록을 버튼을 눌렀을 때 나타나는 질문등록 폼이다.

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

질문등록

제목

두 번째

내용

두 번째

저장하기

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

모델 수정하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 이벤트 핸들러의 정책

- ▶ url매핑에 등록한 `views.question_modify` 함수를 추가하자
- ▶ 수정 버튼을 누르게 되면, 이벤트가 발생해서 이벤트핸들러인 `question_modify()` 함수가 수행되고, 이 함수 안에서 다양한 정책을 수행한다.
- ▶ `post()` 방식의 경우, 즉, 질문을 실제 수정하는 경우, db에 수정 사항을 저장하는 `save()` 함수가 수행되어야 한다.
- ▶ 즉 질문수정 버튼을 누르게 되면 이벤트가 발생하여, 수정할 질문의 제목과 내용을 db로 부터 가져오는 `get()` 작업을 먼저 수행해야 한다,
- ▶ 그리고 질문 수정 폼으로 화면 전환하여, 질문 제목과 질문 내용을 db로부터 가져와서 보여주는 `render`작업을 진행해야 한다.

```
def question_modify(request, question_id):
```

```
    """
```

```
    pybo 질문 수정
```

```
    """
```

```
    question = get_object_or_404(Question, pk=question_id)
```

```
    if request.user != question.author:
        messages.error(request, '수정권한이 없습니다')
        return redirect('pybo:detail', question_id=question.id)
```

```
    if request == "POST":
        form=QuestionForm(request.POST, instance=question)
        if form.is_valid():
            question=form.save(commit=False)
            question.author=request.user
            question.modify_date=timezone.now()
            question.save()
            return redirect('pybo:detail', question_id=question.id)
```

```
    else:
        form=QuestionForm(instance=question)
        context={'form': form}
        return render(request, 'pybo/question_form.html', context)
```

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

모델 수정하기

01단계 이벤트와 이벤트 핸들러의 실행

- ▶ 아래와 같이 수정 버튼이 생성된 것을 볼 수 있다.
 - ▶ 그 다음 수정 버튼을 누르면 `question_modify()` 함수가 호출되고, 정책에 따라 form 화면을 띄우고 수정하고자 하는 질문 제목과 질문 내용을 폼에 채운다
 - ▶ 이 코드가 그 역할을 한다
- ```
form=QuestionForm(instance=question)
```
- ▶ 그리고 아래와 같이 질문 제목을 수정하고 질문 내용을 수정한다.
  - ▶ 그리고 저장하기 버튼을 누르게 되면 아래 코드가 작동한다,
- ```
form=QuestionForm(request.POST, instance=question)
```
- ▶ 수정된 질문제목과 질문 내용이 POST 방식으로 획득되어, 수정되어야하는 `question`에 업데이트한다.

두 번째

두 번째

수정

질문등록

제목

두 번째 제목 수정

내용

두 번째 내용 수정

저장하기

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

질문 수정 기능 추가하기

01단계 질문 수정 버튼 추가하기

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

```
(... 생략 ...)  
<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 text-left">  
        <div class="mb-2">{{ question.author.username }}</div>  
        <div>{{ question.create_date }}</div>  
      </div>  
    </div>  
    {% if request.user == question.author %}  
    <div class="my-3">  
      <a href="{% url 'pybo:question_modify' question.id %}"  
        class="btn btn-sm btn-outline-secondary">수정</a>  
    </div>  
  </div>  
</div>
```

```
</div>  
{% endif %}  
</div>  
(... 생략 ...)
```

- ▶ 질문 수정 버튼은 로그인한 사용자와 글쓴이가 같은 경우에만 보여야 하므로 `{% if request.user == question.author %}`와 같이 추가했다.

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

질문 수정 기능 추가하기

02단계 질문 수정 버튼의 URL 매핑 추가하기

파일 이름 C:/projects/mysite/pybo/urls.py

```
urlpatterns = [
    (... 생략 ...)
    path('question/modify/<int:question_id>/', views.question_modify,
         name='question_modify'),
]
```

- ▶ {% url 'pybo:question_modify' question.id %} URL이 추가되었으니 pybo/urls.py 파일을 수정하여 URL 매핑을 추가해야 한다.

03단계 질문 수정 함수 추가하기

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

```
from django.contrib import messages
(... 생략 ...)

@login_required(login_url='common:login')
def question_modify(request, question_id):
    """
    pybo 질문 수정
    """
    question = get_object_or_404(Question, pk=question_id)
    if request.user != question.author:
        messages.error(request, '수정 권한이 없습니다')
        return redirect('pybo:detail', question_id=question.id)
    if request.method == "POST":
        form = QuestionForm(request.POST, instance=question)
```

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

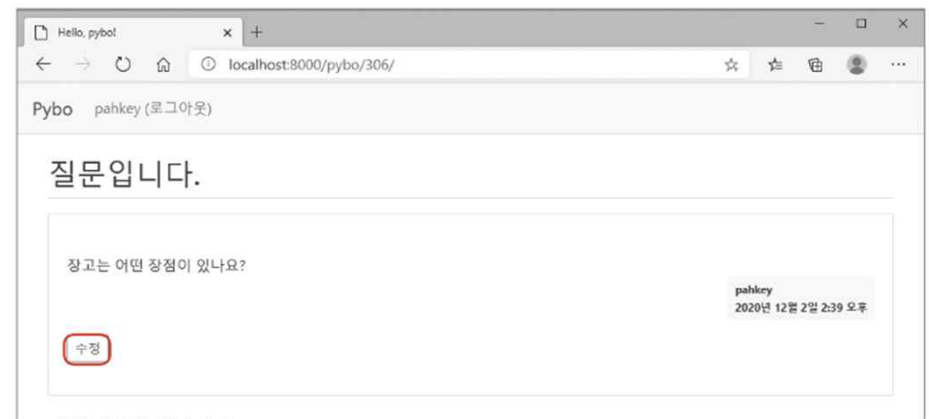
질문 수정 기능 추가하기

03단계 질문 수정 함수 추가하기

```
if form.is_valid():
    question = form.save(commit=False)
    question.author = request.user
    question.modify_date = timezone.now() # 수정일시 저장
    question.save()
    return redirect('pybo:detail', question_id=question.id)
else:
    form = QuestionForm(instance=question)
    context = {'form': form}
    return render(request, 'pybo/question_form.html', context)
```

- ▶ 질문 상세 화면에서 <수정>을 누르면 /pybo/question/modify/2/ 페이지가 GET방식으로 호출되어 질문 수정 화면이 나타나고, 질문 수정 화면에서 <저장하기>를 누르면 /pybo/question/modify/2/ 페이지가 POST방식으로 호출되어 데이터 수정이 이뤄진다.

04단계 질문 수정 확인하기



수정 기능 확인

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

질문 삭제 기능 추가하기

01단계 질문 삭제 버튼 추가하기

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

```
(... 생략 ...)
{% if request.user == question.author %}
<div class="my-3">
  <a href="{% url 'pybo:question_modify' question.id %}"
    class="btn btn-sm btn-outline-secondary">수정</a>
  <a href="#" class="delete btn btn-sm btn-outline-secondary"
    data-uri="{% url 'pybo:question_delete' question.id %}">삭제</a>
</div>
{% endif %}
(... 생략 ...)
```

- ▶ <삭제> 버튼은 <수정> 버튼과는 달리 href 속성값을 “#”로 설정했다.

02단계 질문 삭제 버튼에 jQuery 사용하기

삭제 기능을 위한 제이쿼리 코드

```
<script type='text/javascript'>
$(document).ready(function(){
  $(".delete").on('click', function() {
    if(confirm("정말로 삭제하시겠습니까?")) {
      location.href = $(this).data('uri');
    }
  });
});
```

- ▶ <삭제> 버튼을 누르면 확인 창이 나타나고, <확인> 버튼을 누르면 data-uri 속성값으로 URL이 호출된다. <취소> 버튼을 누르면 아무 일도 발생하지 않는다.

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

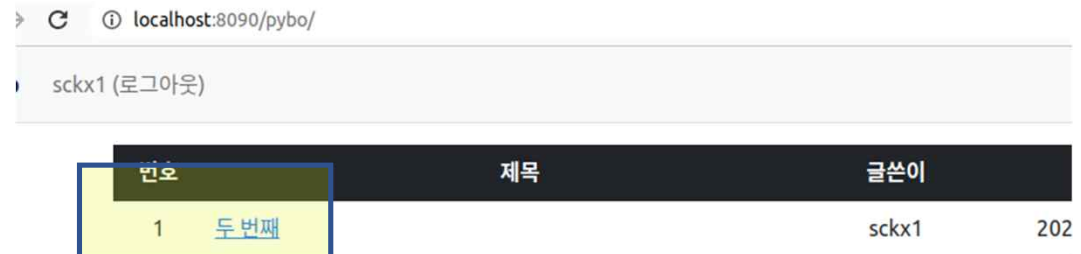
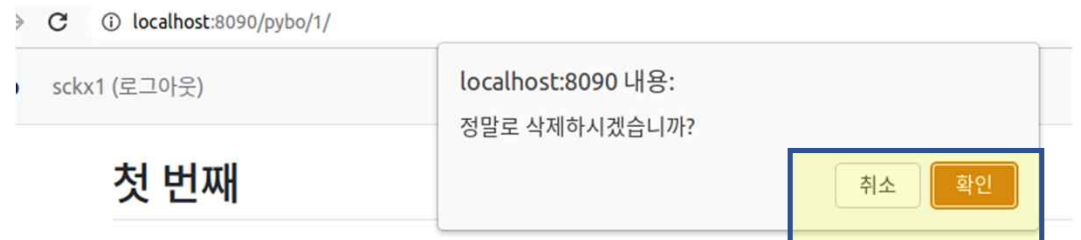
질문 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 질문 삭제 확인 팝업창

- ▶ 질문 삭제는 정말로 질문을 삭제할 것인지 팝업창을 띄워서 사용자에게 한 번 더 확인하는 작업이 필요하다.
- ▶ delete를 수행하기 위해 띄우는 팝업창은 자바스크립트로 되어 있고, 본 과정에서는 그냥 복사해서 붙여넣기 형태로 사용한다.
- ▶ 따로 공부하는 과정은 없다.

- ▶ 확인 버튼을 누르게 되면, 첫 번째 질문이 삭제되고, 질문 리스트 화면이 render 된다.



03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

질문 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 질문 삭제 확인 팝업창

- ▶ 교재의 p180 자바 스크립트로 코딩하고 난 후, 삭제 버튼을 누르면, "삭제하시겠습니까?" 라는 팝업창이 뜨지 않는다.
- ▶ 교재의 p180 자바 스크립트 (작동하지 않음).

```
{% block script %}
<script type='text/javascript'>
    $(document).ready(function () {
        $(".delete").on('click', function () {
            if(confirm("정말로 삭제하시겠습니까?")) {
                location.href = $(this).data('uri');
            }
        });
    });
</script>
{% endblock %}
```

▶ 교재 깃허브의 자바 스크립트 (작동함).

```
82 + {% endblock %}
83 + {% block script %}
84 + <script type='text/javascript'>
85 + const delete_elements =
      document.getElementsByClassName("delete");
86 + Array.from(delete_elements).forEach(function(element) {
87 +     element.addEventListener('click', function() {
88 +         if(confirm("정말로 삭제하시겠습니까?")) {
89 +             location.href = this.dataset.uri;
90 +         };
91 +     });
92 + });
93 + </script>
94 + {% endblock %}
```

03-9 게시물 수정 & 삭제 기능 추가하기

Do it!
실습

질문 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

01단계 질문 삭제 확인 팝업창

- ▶ 수정하고 나서 삭제 버튼을 누르면, 삭제 하십니까 라는 팝업창이 뜬다.

localhost:8000/pybo/2/#

sckx1 (로그아웃)

두 번째

두 번째

수정 삭제

localhost:8000 내용:
정말로 삭제하시겠습니까?

취소

확인

- ▶ 삭제가 잘 이뤄진 화면, 즉 질문을 삭제하고 난 후에는 질문 리스트로 redirect 해서 질문 리스트에서 질문이 삭제 된 것을 확인하면 된다.
- ▶ render는 form을 viewing 할 때 사용하고, redirect는 url로 페이지 이동할 때 사용한다.

Hello, pybo!

Hello, pybo!

localhost:8000/pybo/

Pybo sckx1 (로그아웃)

번호

제목

질문이 없습니다.

이전 1 다음

질문 등록하기

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

질문 삭제 기능 추가하기

03단계 jQuery 실행을 위해 pybo/base.html 파일 수정하기

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

```
(... 생략 ...)  
<!-- jQuery JS -->  
<script src="{% static 'jquery-3.4.1.min.js' %}"></script>  
<!-- Bootstrap JS -->  
<script src="{% static 'bootstrap.min.js' %}"></script>  
<!-- 자바스크립트 Start -->  
{% block script %}  
{% endblock %}  
<!-- 자바스크립트 End -->  
</body>  
</html>
```

- ▶ 제이쿼리를 이용하여 자바스크립트를 작성하기 위해서는 제이쿼리 라이브러리 파일을 먼저 로드해야 한다.

04단계 질문 템플릿에 삭제 알림 창 기능 추가하기

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

```
(... 생략 ...)  
{% endblock %}  
{% block script %}  
<script type='text/javascript'>  
$(document).ready(function(){  
    $(".delete").on('click', function() {  
        if(confirm("정말로 삭제하시겠습니까?")) {  
            location.href = $(this).data('uri');  
        }  
    });  
});  
</script>  
{% endblock %}
```

맨 아래에 추가

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

질문 삭제 기능 추가하기

05단계 질문 삭제 URL 매핑 추가하기

파일 이름 C:/projects/mysite/pybo/urls.py

```
urlpatterns = [
    (... 생략 ...)
    path('question/delete/<int:question_id>/', views.question_delete,
         name='question_delete'),
]
```

- ▶ data-uri 속성에 {% url 'pybo:question_delete' question.id %}이 추가되었으므로 pybo/urls.py 파일에 URL 매핑을 추가해야 한다.

06단계 질문 삭제 함수 추가하기

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

```
(... 생략 ...)
@login_required(login_url='common:login')
def question_delete(request, question_id):
    """
    pybo 질문 삭제
    """
    question = get_object_or_404(Question, pk=question_id)
    if request.user != question.author:
        messages.error(request, '삭제권한이 없습니다')
        return redirect('pybo:detail', question_id=question.id)
    question.delete()
    return redirect('pybo:index')
```

- ▶ question_delete 함수 역시 로그인이 필요하므로 @login_required 애너테이션을 적용하고 로그인한 사용자와 글쓴이가 동일한 경우에만 삭제할 수 있도록 했다.

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

답변 수정 & 삭제 기능 추가하기

01단계 답변 수정 버튼 추가하기

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

```
(... 생략 ...)  
{% for answer in question.answer_set.all %}  
<div class="card my-3">  
  <div class="card-body">  
    (... 생략 ...)  
    {% if request.user == answer.author %}  
      <div class="my-3">  
        <a href="{% url 'pybo:answer_modify' answer.id %}"  
          class="btn btn-sm btn-outline-secondary">수정</a>  
      </div>  
    {% endif %}  
  </div>  
</div>  
{% endfor %}  
(... 생략 ...)
```

card-body 안에 작성!

02단계 답변 수정 URL 매핑 추가하기

파일 이름 C:/projects/mysite/pybo/urls.py

```
urlpatterns = [  
    (... 생략 ...)  
    path('answer/modify/<int:answer_id>/', views.answer_modify, name='answer_modify'),  
]
```

- ▶ {% url 'pybo:answer_modify' answer.id %}가 추가되었으므로 pybo/urls.py 파일에 URL 매핑을 추가해야 한다.

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

답변 수정 & 삭제 기능 추가하기

03단계 답변 수정 함수 추가하기

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

```
(... 생략 ...)
from .models import Question, Answer
(... 생략 ...)
@login_required(login_url='common:login')
def answer_modify(request, answer_id):
    """
    pybo 답변 수정
    """
    answer = get_object_or_404(Answer, pk=answer_id)
    if request.user != answer.author:
        messages.error(request, '수정 권한이 없습니다')
        return redirect('pybo:detail', question_id=answer.question.id)

    if request.method == "POST":
        form = AnswerForm(request.POST, instance=answer)
```

```
        if form.is_valid():
            answer = form.save(commit=False)
            answer.author = request.user
            answer.modify_date = timezone.now()
            answer.save()
            return redirect('pybo:detail', question_id=answer.question.id)
        else:
            form = AnswerForm(instance=answer)
            context = {'answer': answer, 'form': form}
            return render(request, 'pybo/answer_form.html', context)
```

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

답변 수정 & 삭제 기능 추가하기

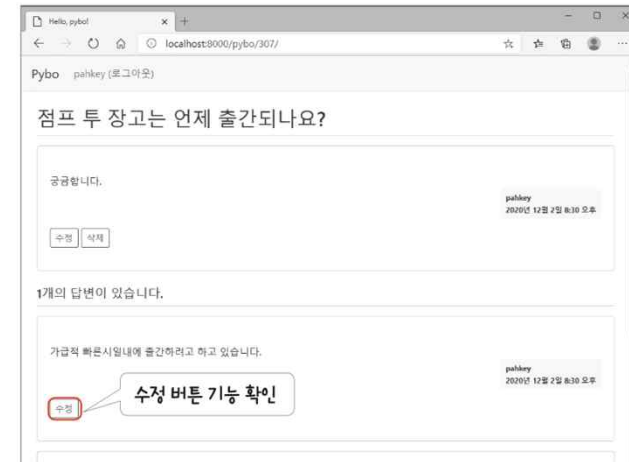
04단계 답변 수정 폼 작성하기

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

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

{% block content %}
<div class="container my-3">
  <form method="post" class="post-form">
    {% csrf_token %}
    {% include "form_errors.html" %}
    <div class="form-group">
      <label for="content">답변내용</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 %}
```

- ▶ 답변 수정 기능도 질문 수정 기능과 마찬가지로 답변 등록 사용자와 로그인 사용자가 동일할 때만 <수정> 버튼이 나타난다.



수정 기능 확인

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

답변 수정 & 삭제 기능 추가하기

05단계 답변 삭제 버튼 추가하기

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

```
(... 생략 ...)
{% for answer in question.answer_set.all %}
<div class="card my-3">
  <div class="card-body">
    (... 생략 ...)
    {% if request.user == answer.author %}
    <div class="my-3">
      <a href="{% url 'pybo:answer_modify' answer.id %}"
        class="btn btn-sm btn-outline-secondary">수정</a>
      <a href="#" class="delete btn btn-sm btn-outline-secondary"
        data-uri="{% url 'pybo:answer_delete' answer.id %}">삭제</a>
    </div>
    {% endif %}
  </div>
</div>
{% endfor %}
(... 생략 ...)
```

- ▶ 질문의 <삭제> 버튼과 마찬가지로 <삭제> 버튼에 delete 클래스를 적용했으므로 <삭제> 버튼을 누르면 data-uri 속성에 설정한 url이 실행될 것이다.

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

답변 수정 & 삭제 기능 추가하기

06단계 답변 삭제 URL 매핑 추가하기

파일 이름 C:/projects/mysite/pybo/urls.py

```
urlpatterns = [
    (... 생략 ...)
    path('answer/delete/<int:answer_id>', views.answer_delete, name='answer_delete'),
]
```

- ▶ {% url 'pybo:answer_delete' answer.id %}이 추가되었으므로 pybo/urls.py 파일에 URL 매핑을 추가하자.

07단계 답변 삭제 함수 추가하기

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

```
(... 생략 ...)
@login_required(login_url='common:login')
def answer_delete(request, answer_id):
    """
    pybo 답변 삭제
    """
    answer = get_object_or_404(Answer, pk=answer_id)
    if request.user != answer.author:
        messages.error(request, '삭제권한이 없습니다')
    else:
        answer.delete()
    return redirect('pybo:detail', question_id=answer.question.id)
```

03-9 게시물 수정 & 삭제 기능 추가하기

• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

수정일시 표시하기

01단계 작성일시 왼쪽에 수정일시 추가하기

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

```
(... 생략 ...)
<div class="card-body">
  <div class="card-text" style="white-space: pre-line;">{{ question.content }}</div>
  <div class="d-flex justify-content-end">
    {% if question.modify_date %}
    <div class="badge badge-light p-2 text-left mx-3">
      <div class="mb-2">modified at</div>
      <div>{{ question.modify_date }}</div>
    </div>
    {% endif %}
    <div class="badge badge-light p-2 text-left">
      <div class="mb-2">{{ question.author.username }}</div>
      <div>{{ question.create_date }}</div>
    </div>
  </div>
  (... 생략 ...)
</div>
```

삽입 위치 확인

(... 생략 ...)

```
{% 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">
      {% if answer.modify_date %}
      <div class="badge badge-light p-2 text-left mx-3">
        <div class="mb-2">modified at</div>
        <div>{{ answer.modify_date }}</div>
      </div>
      {% endif %}
      <div class="badge badge-light p-2 text-left">
        <div class="mb-2">{{ answer.author.username }}</div>
        <div>{{ answer.create_date }}</div>
      </div>
    </div>
  </div>
  (... 생략 ...)
</div>
```

삽입 위치 확인

03-9 게시물 수정 & 삭제 기능 추가하기

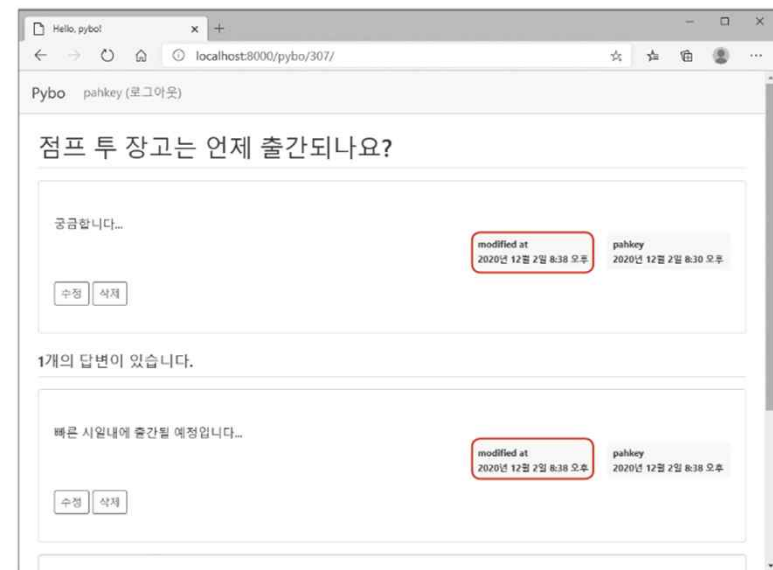
• 완성 소스 github.com/pahkey/djangobook/tree/3-09

Do it!
실습

수정일시 표시하기

```
</div>
</div>
{% endfor %}
(... 생략 ...)
```

- ▶ 이제 질문이나 답변을 수정하면 다음처럼 수정일시가 표시될 것이다.



수정일시 표시 확인

Thank You