5주차 리뷰 특강

▼ PDF 다운받기

[예정]

목차

리뷰 특강 목표

웹서비스의 A-Z 만들기. 굳히기 들어갑니다!

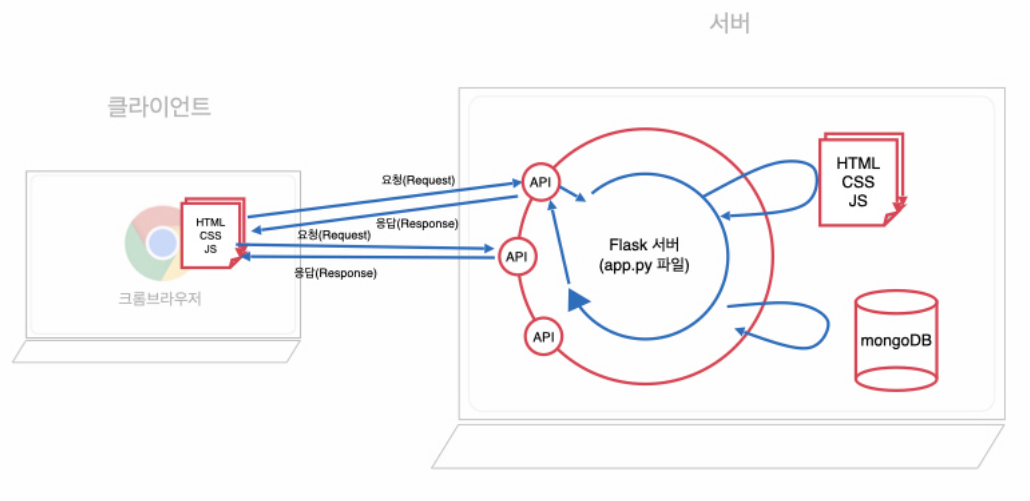
1. **[익숙한 코드로 복습해보기]** 북리뷰 만들어보기
2. **[혼자 다시 해보기]** 코드를 지우고, 다시!
3. **[크롤링 복습해보기]** 데이터를 넣어보자
4. **[미니 프로젝트 해보기]** 이제 혼자 할 수 있다구!

☑️ 체크인

- 딱 세 명만! 기대하는 바 이야기해보기

들어가기 전에

웹서비스의 동작원리를 마음에 담아두기!



오늘 필요할 코드들 모음

▼ 웹스크래핑 기본 코드

```
import requests
from bs4 import BeautifulSoup

url = 'https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20200716'

headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683
data = requests.get(url,headers=headers)

soup = BeautifulSoup(data.text, 'html.parser')
```

▼ pymongo 기본 코드

임포트하기

```
from pymongo import MongoClient
client = MongoClient('localhost', 27017)
db = client.dbreview
```

CRUD 기본 세팅

```
# Create(생성)
user1 = {'name': '론', 'age': 40}
user2 = {'name': '해리', 'age': 40}
db.users.insert_one(user1)
db.users.insert_one(user2)

# Read(조회) - 한 개 값만
user = db.users.find_one({'name': '론'})

# Read(조회) - 여러 값 ( _id 값은 제외하고 출력)
same_ages = list(db.users.find({'age': 40}, {'_id': False}))

# Update(업데이트) - 여러 값
db.people.update_many({'age': 40}, { '$set': {'age': 70}})

# Update(업데이트) - 하나 값
db.users.update_one({'name': '론'}, {'$set': {'age': 116}})

# Delete(삭제)
db.users.delete_one({'name': '론'})
```

▼ GET - API 만들기 / Ajax 요청하기

서버API 코드

```
@app.route('/test', methods=['GET'])
def test_get():
```

```
    title_receive = request.args.get('title_give')
    print(title_receive)
    return jsonify({'result': 'success', 'msg': '이 요청은 GET!'})
```

Ajax 요청 코드

```
$.ajax({
    type: "GET",
    url: "/test?title_give=봄날은간다",
    data: {},
    success: function(response){
       console.log(response)
    }
  })
```

▼ POST - API 만들기 / Ajax 요청하기

서버API 코드

```
@app.route('/test', methods=['POST'])
def test_post():
    title_receive = request.form['title_give']
    print(title_receive)
    return jsonify({'result': 'success', 'msg': '이 요청은 POST!'})
```

Ajax 요청 코드

```
$.ajax({
    type: "POST",
    url: "/test",
    data: { title_give:"봄날은간다" },
    success: function(response){
       console.log(response)
    }
  })
```

🚨 **시작하기 전에! 폴더를 3개 만들고 시작할게요!**
sparta → review → [bookreview1, bookreview2, mycity]

Review - 왔던 길을 함께 짚어보기

🔥 new projct → **sparta → review → bookreview** 를 열어주세요!

▼ 1) 프로젝트 살펴보기 + 세팅하기 + 폴더구조 잡기

- 완성작보기
- 설치 할 패키지

```
flask pymongo
```

▼ 2) 뼈대 코드 붙여넣기

▼ [💻 코드 - `index.html`]

```
<!DOCTYPE html>
<html lang="ko">

    <head>
        <!-- Webpage Title -->
```

```
        <title>모두의 책리뷰 | 스파르타코딩클럽</title>

        <!-- 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" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
              integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
              crossorigin="anonymous">

        <!-- JS -->
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
                integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
                crossorigin="anonymous"></script>

        <!-- 구글폰트 -->
        <link href="https://fonts.googleapis.com/css?family=Do+Hyeon&display=swap" rel="stylesheet">

        <script type="text/javascript">

            $(document).ready(function () {
                $("#reviews-box").html("");
                showReview();
            });

            function makeReview() {
                // 1. 제목, 저자, 리뷰 내용을 가져옵니다.
                // 2. 제목, 저자, 리뷰 중 하나라도 입력하지 않았을 경우 alert를 띄웁니다.
                // 3. POST /review 에 저장을 요청합니다.
                $.ajax({
                    type: "POST",
                    url: "/review",
                    data: {},
                    success: function (response) {
                        if (response["result"] == "success") {
                            alert(response["msg"]);
                            window.location.reload();
                        }
                    }
                })
            }

            function showReview() {
                // 1. 리뷰 목록을 서버에 요청하기
                // 2. 요청 성공 여부 확인하기
                // 3. 요청 성공했을 때 리뷰를 올바르게 화면에 나타내기
                $.ajax({
                    type: "GET",
                    url: "/review",
                    data: {},
                    success: function (response) {
                        if (response["result"] == "success") {
                            alert(response["msg"]);
                        } else {
                            alert("리뷰를 받아오지 못했습니다");
                        }
                    }
                })
            }

            function validateLength(obj) {
                let content = $(obj).val();
                if (content.length > 140) {
                    alert("리뷰는 140자까지 기록할 수 있습니다.");
                    $(obj).val(content.substring(0, content.length - 1));
                }
            }
        </script>

        <style type="text/css">
            * {
                font-family: "Do Hyeon", sans-serif;
            }

            h1,
            h5 {
                display: inline;
            }

            .info {
                margin-top: 20px;
                margin-bottom: 20px;
            }

            .review {
```

```
                text-align: center;
            }

            .reviews {
                margin-top: 100px;
            }
        </style>
    </head>

    <body>
        <div class="container">
            <img src="https://previews.123rf.com/images/maxxyustas/maxxyustas1511/maxxyustas151100002/47858355-educatio
                 class="img-fluid" alt="Responsive image">
            <div class="info">
                <h1>읽은 책에 대해 말씀해주세요.</h1>
                <p>다른 사람을 위해 리뷰를 남겨주세요! 다 같이 좋은 책을 읽는다면 다 함께 행복해질 수 있지 않을까요?</p>
                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                        <span class="input-group-text">제목</span>
                    </div>
                    <input type="text" class="form-control" id="title">
                </div>
                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                        <span class="input-group-text">저자</span>
                    </div>
                    <input type="text" class="form-control" id="author">
                </div>
                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                        <span class="input-group-text">리뷰</span>
                    </div>
                    <textarea class="form-control" id="bookReview"
                              cols="30"
                              rows="5" placeholder="140자까지 입력할 수 있습니다." onkeyup="validateLength(this)"></textarea>
                </div>
                <div class="review">
                    <button onclick="makeReview()" type="button" class="btn btn-primary">리뷰 작성하기</button>
                </div>
            </div>
            <div class="reviews">
                <table class="table">
                    <thead>
                    <tr>
                        <th scope="col">제목</th>
                        <th scope="col">저자</th>
                        <th scope="col">리뷰</th>
                    </tr>
                    </thead>
                    <tbody id="reviews-box">
                    <tr>
                        <td>왕초보 8주 코딩</td>
                        <td>김르탄</td>
                        <td>역시 왕초보 코딩교육의 명가답군요. 따라하다보니 눈 깜짝할 사이에 8주가 지났습니다.</td>
                    </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </body>

</html>
```

▼ [💻 코드 `app.py`]

```
from flask import Flask, render_template, jsonify, request
import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient  # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)

app = Flask(__name__)

client = MongoClient('localhost', 27017)  # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbreview # 'dbreview'라는 이름의 db를 만들거나 사용합니다.


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/memo', methods=['POST'])
def post_article():
    # 1. 클라이언트로부터 데이터를 받기
```

```
    # 2. meta tag를 스크래핑하기
    # 3. mongoDB에 데이터 넣기
    return jsonify({'result': 'success', 'msg': 'POST 연결되었습니다!'})


@app.route('/memo', methods=['GET'])
def read_articles():
    # 1. mongoDB에서 _id 값을 제외한 모든 데이터 조회해오기(Read)
    # 2. articles라는 키 값으로 articles 정보 보내주기
    return jsonify({'result': 'success', 'msg': 'GET 연결되었습니다!'})


if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

▼ 3) API 만들고 사용하기 - 제목, 저자, 리뷰 정보 저장하기(Create → **POST**)

▼ 1. 클라이언트와 서버 확인하기

- 여기서는 적혀 있는 쌍으로 되어있는 서버-클라이언트 코드를 확인하고 갈게요.
- 분홍 형광펜 부분이 서로 어떻게 매칭되는지 확인해보세요!

만들어져 있는 API 정보

1. 요청 정보 : 요청 URL= `/review` , 요청 방식 = `POST`
2. 서버가 제공할 기능 : 클라이언트에게 정해진 메시지를 보냄
3. 응답 데이터 : (JSON 형식) 'result'= 'success', 'msg'= '리뷰가 성공적으로 작성되었습니다.'

[서버 코드 - `app.py`]

```
## API 역할을 하는 부분
@app.route('/review', methods=['POST'])
def write_review():
  # 1. 클라이언트가 준 title, author, review 가져오기.
  # 2. DB에 정보 삽입하기
  # 3. 성공 여부 & 성공 메시지 반환하기
    return jsonify({'result': 'success', 'msg': '리뷰가 성공적으로 작성되었습니다.'})
```

[클라이언트 코드 - `index.html`]

```
function makeReview() {
    // 1. 제목, 저자, 리뷰 내용을 가져옵니다.
    // 2. 제목, 저자, 리뷰 중 하나라도 입력하지 않았을 경우 alert를 띄웁니다.
    // 3. POST /review 에 저장을 요청합니다.
    $.ajax({
        type: "POST",
        url:  "/review",
        data: { },
        success: function (response) {
            if (response["result"] == "success") {
                alert(response["msg"] );
                window.location.reload();
            }
        }
    })
}
```

동작 테스트
'리뷰 시작하기' 버튼을 눌렀을 때, '리뷰가 성공적으로 작성되었습니다.' 라는 내용의 alert창이 뜨면 클라이언트 코드와 서버 코드가 연결 되어있는 것입니다.

▼ 2. 서버부터 만들기

API 는 약속이라고 했습니다. API를 먼저 만들어보죠!

리뷰를 작성하기 위해 필요한 정보는 다음 세 가지 입니다.
- 제목(title)
- 저자(author)
- 리뷰(review)

따라서 API 기능은 다음 세 단계로 구성되어야 합니다.
1. 클라이언트가 준 title, author, review 가져오기.
2. DB에 정보 삽입하기
3. 성공 여부 & 성공 메시지 반환하기

정리하면, **만들 API 정보**는 아래와 같습니다.

A. 요청 정보
- 요청 URL= `/review` , 요청 방식 = `POST`
- 요청 데이터 : 제목(title), 저자(author), 리뷰(review)

B. 서버가 제공할 기능 : 클라이언트에게 보낸 요청 데이터를 데이터베이스에 생성(Create)하고, 저장이 성공했다고 응답 ㅗ데이터를 보냄

C**. 응답 데이터 :** (JSON 형식) 'result'= 'success', 'msg'= '리뷰가 성공적으로 작성되었습니다.'

```
@app.route('/review', methods=['POST'])
def write_review():
    # title_receive로 클라이언트가 준 title 가져오기
    title_receive = request.form['title_give']
    # author_receive로 클라이언트가 준 author 가져오기
    author_receive = request.form['author_give']
    # review_receive로 클라이언트가 준 review 가져오기
    review_receive = request.form['review_give']

    # DB에 삽입할 review 만들기
    review = {
        'title': title_receive,
        'author': author_receive,
        'review': review_receive
    }
    # reviews에 review 저장하기
    db.reviews.insert_one(review)
    # 성공 여부 & 성공 메시지 반환
    return jsonify({'result': 'success', 'msg': '리뷰가 성공적으로 작성되었습니다.'})
```

▼ 3. 클라이언트 만들기

API 는 약속이라고 했습니다. API를 사용할 클라이언트를 만들어보죠!

리뷰를 작성하기 위해 필요한 정보는 다음 세 가지 입니다.
- 제목(title)
- 저자(author)
- 리뷰(review)

따라서 클라이언트 코드는 다음 세 단계로 구성되어야 합니다.
1. input에서 title, author, review 가져오기
2. 입력값이 하나라도 없을 때 alert 띄우기.
3. Ajax로 서버에 저장 요청하고, 화면 다시 로딩하기

사용할 API 정보

A. 요청 정보
- 요청 URL= `/review` , 요청 방식 = `POST`
- 요청 데이터 : 제목(title), 저자(author), 리뷰(review)

B. 서버가 제공할 기능 : 클라이언트에게 보낸 요청 데이터를 데이터베이스에 생성(Create)하고, 저장이 성공했다고 응답 데이터를 보냄

C. 응답 데이터 : (JSON 형식) 'result'= 'success', 'msg'= '리뷰가 성공적으로 작성되었습니다.'

```
function makeReview() {
    // 1. 화면에 입력어 있는 제목, 저자, 리뷰 내용을 가져옵니다.
    let title = $("#title").val();
    let author = $("#author").val();
    let review = $("#bookReview").val();

    // 2. 제목, 저자, 리뷰 중 하나라도 입력하지 않았을 경우 alert를 띄웁니다.
    if (title == "") {
        alert("제목을 입력해주세요");
        $("#title").focus();
        return;
    } else if (author == "") {
        alert("저자를 입력해주세요");
        $("#author").focus();
        return;
    } else if (review == "") {
        alert("리뷰를 입력해주세요");
        $("#bookReview").focus();
        return;
    }

    // 3. POST /review 에 저장(Create)을 요청합니다.
    $.ajax({
        type: "POST",
        url: "/review",
        data: { title_give: title, author_give: author, review_give: review },
        success: function (response) {
            if (response["result"] == "success") {
                alert(response["msg"]);
                window.location.reload();
            }
        }
    })
}
```

▼ 4. 완성 확인하기

동작 테스트
제목, 저자, 리뷰를 작성하고 '리뷰 작성하기' 버튼을 눌렀을 때,
'리뷰가 성공적으로 작성되었습니다.'라는 alert가 뜨는지 확인합니다.

▼ 4) API 만들고 사용하기 - 저장된 리뷰를 화면에 보여주기(Read → **GET**)

▼ 1. 클라이언트와 서버 확인하기

- 여기서는 미리 적혀 있는 쌍으로 되어있는 서버-클라이언트 코드를 확인하고 갈게요.
- 분홍 형광펜 부분이 서로 어떻게 매칭되는지 확인해보세요!

만들어져 있는 API 정보

1. 요청 정보 : 요청 URL= `/review` , 요청 방식 = `GET`
2. 서버가 제공할 기능 : 클라이언트에게 정해진 메시지를 보냄
3. 응답 데이터 : (JSON 형식) 'result'= 'success'

[서버 코드 - `app.py`]

```
@app.route('/review', methods=['GET'])
def read_reviews():
    # 1. 모든 reviews의 문서를 가져온 후 list로 변환합니다.
    # 2. 성공 메시지와 함께 리뷰를 보냅니다.
    return jsonify({'result': 'success'})
```

[클라이언트 코드 - `index.html`]

```
function showReview() {
    // 1. 리뷰 목록을 서버에 요청하기
    // 2. 요청 성공 여부 확인하기
    // 3. 요청 성공했을 때 리뷰를 올바르게 화면에 나타내기
    $.ajax({
        type: "GET",
        url: "/review",
        data: {},
        success: function (response) {
            if (response['result'] == 'success') {
                alert('리뷰를 받아왔습니다.');
                // 2. 성공했을 때 리뷰를 올바르게 화면에 나타내기
            } else {
                alert('리뷰를 받아오지 못했습니다');
            }
        }
    })
}
```

동작 테스트
화면을 새로고침 했을 때, '리뷰를 받아왔습니다.' 라는 내용의 alert창이 뜨면
클라이언트 코드와 서버 코드가 연결 되어있는 것입니다.

▼ 2. 서버부터 만들기

API 는 약속이라고 했습니다. API를 먼저 만들어보죠!

API 기능은 다음 단계로 구성되어야 합니다.
1. DB에서 리뷰 정보 모두 가져오기
2. 성공 여부 & 리뷰 목록 반환하기

정리하면, **만들 API 정보**는 아래와 같습니다.

A. 요청 정보
- 요청 URL= `/review` , 요청 방식 = `GET`
- 요청 데이터 : 없음

B. 서버가 제공할 기능 : 데이터베이스에 리뷰 정보를 조회(Read)하고, 성공 메시지와 리뷰 정보를 응답 데이터를 보냄

C. 응답 데이터 : (JSON 형식) 'result'= 'success', 'reviews'= 리뷰리스트

```
@app.route('/review', methods=['GET'])
def read_reviews():
    # 1. DB에서 리뷰 정보 모두 가져오기
    reviews = list(db.reviews.find({}, {'_id': 0}))
    # 2. 성공 여부 & 리뷰 목록 반환하기
    return jsonify({'result': 'success', 'reviews': reviews})
```

▼ 3. 클라이언트 만들기

API 는 약속이라고 했습니다. API를 사용할 클라이언트를 만들어보죠!

리뷰를 작성하기 위해 필요한 정보는 다음 세 가지 입니다.
- 제목(title)
- 저자(author)
- 리뷰(review)

따라서 클라이언트 코드는 다음 세 단계로 구성되어야 합니다.
1. 리뷰 목록을 서버에 요청하기
2. 요청 성공 여부 확인하기
3. 요청 성공했을 때 리뷰를 올바르게 화면에 나타내기

사용할 API 정보는 아래와 같습니다.

A. 요청 정보
- 요청 URL= `/review` , 요청 방식 = `GET`
- 요청 데이터 : 없음

B. 서버가 제공할 기능 : 데이터베이스에 리뷰 정보를 조회(Read)하고, 성공 메시지와 리뷰 정보를 응답 데이터를 보냄

C. 응답 데이터 : (JSON 형식) 'result'= 'success', 'reviews'= 리뷰리스트

```
function showReview() {
    // 1. 리뷰 목록을 서버에 요청하기
    $.ajax({
        type: "GET",
        url: "/review",
        data: {},
        success: function (response) {
            // 2. 요청 성공 여부 확인하기
            if (response["result"] == "success") {
                let reviews = response["reviews"];
                // 3. 요청 성공했을 때 리뷰를 올바르게 화면에 나타내기
                for (let i = 0; i < reviews.length; i++) {
                    makeCard(reviews[i]["title"], reviews[i]["author"], reviews[i]["review"]);
                }
            } else {
                alert("리뷰를 받아오지 못했습니다");
            }
        }
    })
}


function makeCard(title, author, review) {
    let tempHtml = `<tr>
                        <td>${title}</td>
                        <td>${author}</td>
                        <td>${review}</td>
                    </tr>`;
    $("#reviews-box").append(tempHtml);
}
```

▼ 4. 완성 확인하기

동작 테스트
화면을 새로고침 했을 때, DB에 저장된 리뷰가 화면에 올바르게 나타나는지 확인합니다.

Review - 다시 돌아가서 혼자 와보기

🔥 new projct → **sparta → review → bookreview2** 를 열어주세요!

▼ 5) 프로젝트 살펴보기 + 세팅하기 + 폴더구조 잡기

- 완성작보기
- 설치 할 패키지

```
flask pymongo
```

▼ 6) 뼈대 코드 붙여넣기

▼ [💻 코드 - `index.html`]

```
<!DOCTYPE html>
<html lang="ko">

    <head>
        <!-- Webpage Title -->
        <title>모두의 책리뷰 | 스파르타코딩클럽</title>

        <!-- 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" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
              integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
              crossorigin="anonymous">

        <!-- JS -->
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
                integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
                crossorigin="anonymous"></script>

        <!-- 구글폰트 -->
        <link href="https://fonts.googleapis.com/css?family=Do+Hyeon&display=swap" rel="stylesheet">

        <script type="text/javascript">

            $(document).ready(function () {
                $("#reviews-box").html("");
                showReview();
            });

            function makeReview() {
                // 1. 제목, 저자, 리뷰 내용을 가져옵니다.
                // 2. 제목, 저자, 리뷰 중 하나라도 입력하지 않았을 경우 alert를 띄웁니다.
                // 3. POST /review 에 저장을 요청합니다.
                $.ajax({
                    type: "POST",
                    url: "/review",
                    data: {},
                    success: function (response) {
                        if (response["result"] == "success") {
                            alert(response["msg"]);
                            window.location.reload();
                        }
                    }
                })
            }

            function showReview() {
                // 1. 리뷰 목록을 서버에 요청하기
                // 2. 요청 성공 여부 확인하기
                // 3. 요청 성공했을 때 리뷰를 올바르게 화면에 나타내기
                $.ajax({
                    type: "GET",
                    url: "/review",
                    data: {},
                    success: function (response) {
                        if (response["result"] == "success") {
                            alert(response["msg"]);
                        } else {
                            alert("리뷰를 받아오지 못했습니다");
                        }
                    }
                })
```

```
            }

            function validateLength(obj) {
                let content = $(obj).val();
                if (content.length > 140) {
                    alert("리뷰는 140자까지 기록할 수 있습니다.");
                    $(obj).val(content.substring(0, content.length - 1));
                }
            }
        </script>

        <style type="text/css">
            * {
                font-family: "Do Hyeon", sans-serif;
            }

            h1,
            h5 {
                display: inline;
            }

            .info {
                margin-top: 20px;
                margin-bottom: 20px;
            }

            .review {
                text-align: center;
            }

            .reviews {
                margin-top: 100px;
            }
        </style>
    </head>

    <body>
        <div class="container">
            <img src="https://previews.123rf.com/images/maxxyustas/maxxyustas1511/maxxyustas151100002/47858355-educatio
                 class="img-fluid" alt="Responsive image">
            <div class="info">
                <h1>읽은 책에 대해 말씀해주세요.</h1>
                <p>다른 사람을 위해 리뷰를 남겨주세요! 다 같이 좋은 책을 읽는다면 다 함께 행복해질 수 있지 않을까요?</p>
                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                        <span class="input-group-text">제목</span>
                    </div>
                    <input type="text" class="form-control" id="title">
                </div>
                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                        <span class="input-group-text">저자</span>
                    </div>
                    <input type="text" class="form-control" id="author">
                </div>
                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                        <span class="input-group-text">리뷰</span>
                    </div>
                    <textarea class="form-control" id="bookReview"
                              cols="30"
                              rows="5" placeholder="140자까지 입력할 수 있습니다." onkeyup="validateLength(this)"></textarea>
                </div>
                <div class="review">
                    <button onclick="makeReview()" type="button" class="btn btn-primary">리뷰 작성하기</button>
                </div>
            </div>
            <div class="reviews">
                <table class="table">
                    <thead>
                    <tr>
                        <th scope="col">제목</th>
                        <th scope="col">저자</th>
                        <th scope="col">리뷰</th>
                    </tr>
                    </thead>
                    <tbody id="reviews-box">
                    <tr>
                        <td>왕초보 8주 코딩</td>
                        <td>김르탄</td>
                        <td>역시 왕초보 코딩교육의 명가답군요. 따라하다보니 눈 깜짝할 사이에 8주가 지났습니다.</td>
                    </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </body>
```

```
</html>
```

▼ [💻 코드 `app.py`]

```
from flask import Flask, render_template, jsonify, request
import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient  # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)

app = Flask(__name__)

client = MongoClient('localhost', 27017)  # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbreview # 'dbreview'라는 이름의 db를 만들거나 사용합니다.


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/memo', methods=['POST'])
def post_article():
    # 1. 클라이언트로부터 데이터를 받기
    # 2. meta tag를 스크래핑하기
    # 3. mongoDB에 데이터 넣기
    return jsonify({'result': 'success', 'msg': 'POST 연결되었습니다!'})


@app.route('/memo', methods=['GET'])
def read_articles():
    # 1. mongoDB에서 _id 값을 제외한 모든 데이터 조회해오기(Read)
    # 2. articles라는 키 값으로 articles 정보 보내주기
    return jsonify({'result': 'success', 'msg': 'GET 연결되었습니다!'})


if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

▼ 7) API 만들고 사용하기 - 제목, 저자, 리뷰 정보 저장하기(Create → **POST**)

> 🤓 직접 수행해보세요!
> 1) 서버-클라이언트 연결 확인
> 2) 서버
> 3) 클라이언트
> 4) 완성 확인!

▼ 8) API 만들고 사용하기 - 저장된 리뷰를 화면에 보여주기(Read → **GET**)

> 🤓 직접 수행해보세요!
> 1) 서버-클라이언트 연결 확인
> 2) 서버
> 3) 클라이언트
> 4) 완성 확인!

새로운 프로젝트를 해보기!

> 🔥 new projct → **sparta → review → mytravel** 를 열어주세요!

▼ 9) 프로젝트 살펴보기 + 세팅하기 + 폴더구조 잡기

- 완성작 보기
- 설치 할 패키지

```
flask pymongo
```

▼ 10) 뼈대 코드 붙여넣기

▼ [💻 코드 - `index.html`]

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>나홀로여행</title>

    <link href="https://fonts.googleapis.com/css2?family=Poor+Story&display=swap" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.8.0/css/bulma.min.css">
    <script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
    <style>
        * {
            font-family: 'Poor Story', cursive;
        }

        .posting-box {
            max-width: 500px;
            width: 90%;
            margin: 20px auto 20px auto;

            padding: 20px;
            background-color: whitesmoke;
        }

        .mycity > h3 {
            font-weight: bold;
            font-size: 1.5rem;
            margin-bottom: 10px;
        }

        .mycity > h3 > span {
            font-weight: normal;
            font-size: 1.2rem;
        }

        .mycity > p {
            font-size: 1.2rem;
        }
    </style>
    <script>
        $(document).ready(function () {
            show();
        });

        function show() {
            $.ajax({
                type: "GET",
                url: "/show",
                data: {},
                success: function (response) {
                    if (response['result'] == 'success') {
                        alert(response['msg'])
                    }
                }
            })
        }

        function save() {
            $.ajax({
                type: "POST",
                url: "/save",
                data: {},
                success: function (response) {
                    if (response['result'] == 'success') {
                        alert(response['msg'])
                        window.location.reload();
                    }
                }
            })
        }
    </script>
</head>
<body>
<section class="hero is-danger">
    <div class="hero-body has-text-centered">
        <div class="container">
```

```
            <h1 class="title">
                언젠가는 가고 말거야!
            </h1>
            <h2 class="subtitle">
                내가 가고 싶은 여행지 모음
            </h2>
        </div>
    </div>
</section>
<div class="posting-box">
    <div class="field">
        <label class="label">가고 싶은 도시</label>
        <div class="control">
            <input id="input-city" class="input" type="text" placeholder="예) 일본 도쿄">
        </div>
    </div>
    <div class="field">
        <label class="label">이미지 주소</label>
        <div class="control">
            <input id="input-image" class="input" type="text" placeholder="이미지 url을 복사해서 넣기">
        </div>
    </div>
    <div class="field">
        <label class="label">언제갈까?</label>
        <div class="control">
            <input id="input-when" class="input" type="text" placeholder="예) 2021.06">
        </div>
    </div>
    <div class="field">
        <label class="label">가고 싶은 이유</label>
        <div class="control">
            <textarea id="input-memo" class="textarea" placeholder="도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라."></textar
        </div>
    </div>
    <div class="field is-grouped">
        <div class="control">
            <button onclick="save()" class="is-danger button is-link">기록해두기</button>
        </div>
    </div>
</div>
<div class='columns is-multiline' id="city-box">
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
```

```
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
</div>
</body>
</html>
```

▼ [💻 코드 `app.py`]

```
from flask import Flask, render_template, jsonify, request
from pymongo import MongoClient

app = Flask(__name__)

client = MongoClient('localhost', 27017)
db = client.dbreview


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/save', methods=['POST'])
def post_article():
    return jsonify({'result': 'success', 'msg': 'POST 연결되었습니다!'})


@app.route('/show', methods=['GET'])
def read_articles():
    return jsonify({'result': 'success', 'msg': 'GET 연결되었습니다!'})


if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

▼ 11) 기록해두기API (Create → **POST**)

> 🤓 직접 수행해보세요!
> 1) 서버-클라이언트 연결 확인
> 2) 서버
> 3) 클라이언트
> 4) 완성 확인!

▼ 12) 보여주기API(Read → **GET**)

> 🤓 직접 수행해보세요!
> 1) 서버-클라이언트 연결 확인
> 2) 서버
> 3) 클라이언트
> 4) 완성 확인!

▼ 13) 완성코드

▼ [💻 코드 - `index.html`]

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>나홀로여행</title>

    <link href="https://fonts.googleapis.com/css2?family=Poor+Story&display=swap" rel="stylesheet">
```

```
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.8.0/css/bulma.min.css">
    <script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
    <style>
        * {
            font-family: 'Poor Story', cursive;
        }

        .posting-box {
            max-width: 500px;
            width: 90%;
            margin: 20px auto 20px auto;

            padding: 20px;
            background-color: whitesmoke;
        }

        .mycity > h3 {
            font-weight: bold;
            font-size: 1.5rem;
            margin-bottom: 10px;
        }

        .mycity > h3 > span {
            font-weight: normal;
            font-size: 1.2rem;
        }

        .mycity > p {
            font-size: 1.2rem;
        }
    </style>
    <script>
        $(document).ready(function () {
            show();
        });

        function show() {
            $('#city-box').empty();
            $.ajax({
                type: "GET",
                url: "/show",
                data: {},
                success: function (response) {
                    if (response['result'] == 'success') {
                        let mycities = response['mycities'];
                        for (let i = 0; i < mycities.length; i++) {
                            let city = mycities[i]['city']
                            let image = mycities[i]['image']
                            let when = mycities[i]['when']
                            let memo = mycities[i]['memo']

                            let temp_html = `<div class='column is-4'">
                                                <div class="card">
                                                    <div class="card-image">
                                                        <figure class="image is-4by3">
                                                            <img src="${image}"
                                                                 alt="Placeholder image">
                                                        </figure>
                                                    </div>
                                                    <div class="card-content mycity">
                                                        <h3>${city} <span>(예정: ${when})</span></h3>
                                                        <p>${memo}</p>
                                                    </div>
                                                </div>
                                            </div>`
                            $('#city-box').append(temp_html)
                        }
                    }
                }
            })
        }

        function save() {
            let city = $('#input-city').val();
            let image = $('#input-image').val();
            let when = $('#input-when').val();
            let memo = $('#input-memo').val();

            $.ajax({
                type: "POST",
                url: "/save",
                data: {city_give: city, image_give: image, when_give: when, memo_give: memo},
                success: function (response) {
                    if (response['result'] == 'success') {
                        alert(response['msg'])
```

```
                        window.location.reload();
                    }
                }
            })
        }
    </script>
</head>
<body>
<section class="hero is-danger">
    <div class="hero-body has-text-centered">
        <div class="container">
            <h1 class="title">
                언젠가는 가고 말거야!
            </h1>
            <h2 class="subtitle">
                내가 가고 싶은 여행지 모음
            </h2>
        </div>
    </div>
</section>
<div class="posting-box">
    <div class="field">
        <label class="label">가고 싶은 도시</label>
        <div class="control">
            <input id="input-city" class="input" type="text" placeholder="예) 일본 도쿄">
        </div>
    </div>
    <div class="field">
        <label class="label">이미지 주소</label>
        <div class="control">
            <input id="input-image" class="input" type="text" placeholder="이미지 url을 복사해서 넣기">
        </div>
    </div>
    <div class="field">
        <label class="label">언제갈까?</label>
        <div class="control">
            <input id="input-when" class="input" type="text" placeholder="예) 2021.06">
        </div>
    </div>
    <div class="field">
        <label class="label">가고 싶은 이유</label>
        <div class="control">
            <textarea id="input-memo" class="textarea" placeholder="도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라."></textar
        </div>
    </div>
    <div class="field is-grouped">
        <div class="control">
            <button onclick="save()" class="is-danger button is-link">기록해두기</button>
        </div>
    </div>
</div>
<div class='columns is-multiline' id="city-box">
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
```

```
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
    <div class='column is-4'>
        <div class="card">
            <div class="card-image">
                <figure class="image is-4by3">
                    <img src="https://d20aeo683mqd6t.cloudfront.net/ko/articles/title_images/000/038/800/medium/pixta_5
                         alt="Placeholder image">
                </figure>
            </div>
            <div class="card-content mycity">
                <h3>도쿄 <span>(예정: 2020.06)</span></h3>
                <p>도쿄의 벚꽃을 보고 싶다. 마지막으로 간 게 언제였더라.</p>
            </div>
        </div>
    </div>
</div>
</body>
</html>
```

▼ [💻 코드 `app.py`]

```
from flask import Flask, render_template, jsonify, request
from pymongo import MongoClient

app = Flask(__name__)

client = MongoClient('localhost', 27017)
db = client.dbreview


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/save', methods=['POST'])
def post_article():
    city_receive = request.form['city_give']
    image_receive = request.form['image_give']
    when_receive = request.form['when_give']
    memo_receive = request.form['memo_give']

    doc = {
        'city': city_receive,
        'image': image_receive,
        'when': when_receive,
        'memo': memo_receive
    }

    db.mycity.insert_one(doc)

    return jsonify({'result': 'success', 'msg': '저장 완료!'})


@app.route('/show', methods=['GET'])
def read_articles():
    mycities = list(db.mycity.find({},{'_id':False}))
    return jsonify({'result': 'success', 'mycities': mycities})


if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

☑️ 체크 아웃

- 슬랙 thread에 **오늘 내가 배운 것** 한 가지 적기!
  - 예. Ajax로 요청한 데이터 화면에 보여주기