**웹개발 종합반 78기(튜터: 이범규)**

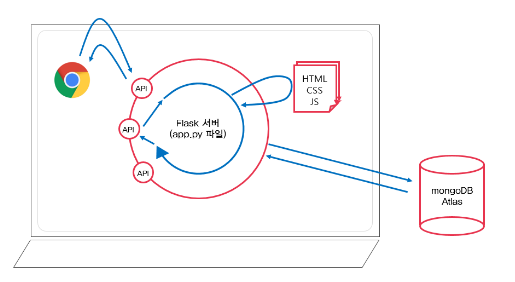
**수업노트 4주차 (22/05/07)**

4주차의 수업이 시작되었다.

| 잠시 수업을 멈추고 스파르타 코딩 클럽을 생각해 본다.  왜 이런 교육과정을 만들게 되었을까?  잠시 구글링… [2년만에 매출액 30배 성장… 하지만 진짜 중요한 수치는 완강률 85%](https://ppss.kr/archives/249092) 그랬구나…사업가였구나.  “**직장인**”을 대상으로, “**코딩교육**”에 집중하고, “**완강율**”에 목메이고,“**취업율**”을 높이는 것으로 피벗을 완성했다는 것을 알게 되었다.  수강생인 내가 교육과정에 집중할 수 있는 것 또한 스파르타코딩클럽 팀이 끌어가고 있는 이런 기업 문화가 바탕이 되고 있구나 라는 생각이 든다.  딴 생각을 멈추고 다시 수업으로 집중해 본다. |
| --- |

**교육과정 리마인드**

본 웹 개발 교육과정의 목표는 웹 서버와 클라우드 DB를 만들고, 외부 웹서버와 연동하는 웹 프로그램을 만드는 것이다.



**프로젝트 폴더 생성**

sparta 폴더 내 bucket, homework, mars, movie, prac 5개 폴더 생성

|  |
| --- |

**Python 프로젝트 생성**

**1. prac 프로젝트 생성**

생성시 위치내 venv 포함여부, 기본디렉터리내 Python38 확인 후 생성!

|  |
| --- |

1.2. 서버 생성용 flask 패키지 설치

파일>설정>프로젝트>Python인터프리터>+>flask 검색 후 패키지 설치

|  |
| --- |

Flask 서버 구동 테스트

App.py 추가 후 다음 코드 저장/실행

| from flask import Flask  app = Flask(\_\_name\_\_)  @app.route('/')  def home():  return 'This is Home!'  if \_\_name\_\_ == '\_\_main\_\_':  app.run('0.0.0.0',port=5000,debug=True)  @app.route('/mypage')  def mypage():  return 'This is MyPage!'  if \_\_name\_\_ == '\_\_main\_\_':  app.run('0.0.0.0',port=5000,debug=True) |
| --- |

서버 접속 확인

2개 페이지 접속 확인!

|  |
| --- |

**Falsk 기본 폴더구조 생성**

* templates: html 페이지들
* static: 이미지 등

Flask 동작 확인

templates\index.html 생성/저장

| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Title</title>  </head>  <body>  <h1>나의 첫 웹페이지!</h1>  <button>버튼을 만들자</button>  </body>  </html> |
| --- |

app.py 수정

templates 내 index.html 내용을 출력!

| from flask import Flask, render\_template  app = Flask(\_\_name\_\_)  @app.route('/')  def home():  return render\_template('index.html')  if \_\_name\_\_ == '\_\_main\_\_':  app.run('0.0.0.0',port=5000,debug=True) |
| --- |

결과는 성공. 안나온다면 뭔가 잘못. slack-FAQ 뒤져보면 됨.

|  |
| --- |

**API 만들기**

GET / POST 방식 학습

GET 보다 POST 방식을 더 많이 쓰게 된다!

index.html

GET, POST 호출 AJAX코드

| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Title</title>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  <script>  function hey() {  $.ajax({  type: "GET",  url: "/test?title\_give=봄날은간다",  data: {},  success: function (response) {  ***console***.log(response)  }  })  $.ajax({  type: "POST",  url: "/test",  data: {title\_give: '봄날은간다'},  success: function (response) {  ***console***.log(response['msg'])  }  })  }  </script>  </head>  <body>  <h1>나의 첫 웹페이지!</h1>  <button onclick="hey()">버튼을 만들자</button>  </body>  </html> |
| --- |

App.py

GET/POST 차례로 호출

| from flask import Flask, render\_template, request, jsonify  app = Flask(\_\_name\_\_)  @app.route('/')  def home():  return render\_template('index.html')  @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!'})  @app.route('/test', methods=['POST'])  def test\_post():  title\_receive = request.form['title\_give']  print(title\_receive)  return jsonify({'result':'success', 'msg': '요청을 잘 받았어요'})  if \_\_name\_\_ == '\_\_main\_\_':  app.run('0.0.0.0',port=5000,debug=True) |
| --- |

결과

|  |
| --- |

**[화성땅 공동구매]-프로젝트 셋팅**

1. **새 프로젝트 생성**

PyCharm구동>새 프로젝트>mars 폴더 지정 후 생성

**2. 기본 폴더/파일 생성**

templates, static 폴더 추가, app.py, index.html 파일 생성

**3. 패키지 설치**

Flask(웹서버), PyMongo, DNSPython(이상 DB) 3개 패키지 추가 (설치방법은 생략)

구성 완료!

**[화성땅 공동구매]-뼈대 준비하기**

App.py

| from flask import Flask, render\_template, request, jsonify  app = Flask(\_\_name\_\_)  @app.route('/')  def home():  return render\_template('index.html')  @app.route("/mars", methods=["POST"])  def web\_mars\_post():  sample\_receive = request.form['sample\_give']  print(sample\_receive)  return jsonify({'msg': 'POST 연결 완료!'})  @app.route("/mars", methods=["GET"])  def web\_mars\_get():  return jsonify({'msg': 'GET 연결 완료!'})  if \_\_name\_\_ == '\_\_main\_\_':  app.run('0.0.0.0', port=5000, debug=True) |
| --- |

index.html

| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  crossorigin="anonymous"></script>  <link href="https://fonts.googleapis.com/css2?family=Gowun+Batang:wght@400;700&display=swap" rel="stylesheet">  <title>선착순 공동구매</title>  <style>  \* {  font-family: 'Gowun Batang', serif;  color: white;  }  body {  background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://cdn.aitimes.com/news/photo/202010/132592\_129694\_3139.jpg');  background-position: center;  background-size: cover;  }  h1 {  font-weight: bold;  }  .order {  width: 500px;  margin: 60px auto 0px auto;  padding-bottom: 60px;  }  .mybtn {  width: 100%;  }  .order > table {  margin: 40px 0;  font-size: 18px;  }  option {  color: black;  }  </style>  <script>  $(***document***).ready(function () {  show\_order();  });  function show\_order() {  $.ajax({  type: 'GET',  url: '/mars',  data: {},  success: function (response) {  alert(response['msg'])  }  });  }  function save\_order() {  $.ajax({  type: 'POST',  url: '/mars',  data: {sample\_give: '데이터전송'},  success: function (response) {  alert(response['msg'])  }  });  }  </script>  </head>  <body>  <div class="mask"></div>  <div class="order">  <h1>화성에 땅 사놓기!</h1>  <h3>가격: 평 당 500원</h3>  <p>  화성에 땅을 사둘 수 있다고?<br/>  앞으로 백년 간 오지 않을 기회. 화성에서 즐기는 노후!  </p>  <div class="order-info">  <div class="input-group mb-3">  <span class="input-group-text">이름</span>  <input id="name" type="text" class="form-control">  </div>  <div class="input-group mb-3">  <span class="input-group-text">주소</span>  <input id="address" type="text" class="form-control">  </div>  <div class="input-group mb-3">  <label class="input-group-text" for="size">평수</label>  <select class="form-select" id="size">  <option selected>-- 주문 평수 --</option>  <option value="10평">10평</option>  <option value="20평">20평</option>  <option value="30평">30평</option>  <option value="40평">40평</option>  <option value="50평">50평</option>  </select>  </div>  <button onclick="save\_order()" type="button" class="btn btn-warning mybtn">주문하기</button>  </div>  <table class="table">  <thead>  <tr>  <th scope="col">이름</th>  <th scope="col">주소</th>  <th scope="col">평수</th>  </tr>  </thead>  <tbody>  <tr>  <td>홍길동</td>  <td>서울시 용산구</td>  <td>20평</td>  </tr>  <tr>  <td>임꺽정</td>  <td>부산시 동구</td>  <td>10평</td>  </tr>  <tr>  <td>세종대왕</td>  <td>세종시 대왕구</td>  <td>30평</td>  </tr>  </tbody>  </table>  </div>  </body>  </html> |
| --- |

MongoDB 접속

|  |
| --- |

[화성땅 공동구매]-POST연습(주문 저장)

주문하기, 주문보기 차례로 구현해 보기

주문하기: POST 방식

주문 보기: GET 방식

| index.html |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  crossorigin="anonymous"></script>  <link href="https://fonts.googleapis.com/css2?family=Gowun+Batang:wght@400;700&display=swap" rel="stylesheet">  <title>선착순 공동구매</title>  <style>  \* {  font-family: 'Gowun Batang', serif;  color: white;  }  body {  background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://cdn.aitimes.com/news/photo/202010/132592\_129694\_3139.jpg');  background-position: center;  background-size: cover;  }  h1 {  font-weight: bold;  }  .order {  width: 500px;  margin: 60px auto 0px auto;  padding-bottom: 60px;  }  .mybtn {  width: 100%;  }  .order > table {  margin: 40px 0;  font-size: 18px;  }  option {  color: black;  }  </style>  <script>  $(***document***).ready(function () {  show\_order();  });  function show\_order() {  $.ajax({  type: 'GET',  url: '/mars',  data: {},  success: function (response) {  // alert(response['msg'])  // console.log(response)  let rows = response['orders']  for (let i = 0; i < rows.length; i++) {  let name = rows[i]['name']  let address = rows[i]['address']  let size = rows[i]['size']  // console.log(name, address, size)  let temp\_html = `<tr>  <td>${name}</td>  <td>${address}</td>  <td>${size}</td>  </tr>`  $('#order-box').append(temp\_html)  }  }  });  }  function save\_order() {  let name = $('#name').val()  let address = $('#address').val()  let size = $('#size').val()  $.ajax({  type: 'POST',  url: '/mars',  data: {name\_give: name, address\_give: address, size\_give: size},  success: function (response) {  alert(response['msg'])  ***window***.location.reload()  }  });  }  </script>  </head>  <body>  <div class="mask"></div>  <div class="order">  <h1>화성에 땅 사놓기!</h1>  <h3>가격: 평 당 500원</h3>  <p>  화성에 땅을 사둘 수 있다고?<br/>  앞으로 백년 간 오지 않을 기회. 화성에서 즐기는 노후!  </p>  <div class="order-info">  <div class="input-group mb-3">  <span class="input-group-text">이름</span>  <input id="name" type="text" class="form-control">  </div>  <div class="input-group mb-3">  <span class="input-group-text">주소</span>  <input id="address" type="text" class="form-control">  </div>  <div class="input-group mb-3">  <label class="input-group-text" for="size">평수</label>  <select class="form-select" id="size">  <option selected>-- 주문 평수 --</option>  <option value="10평">10평</option>  <option value="20평">20평</option>  <option value="30평">30평</option>  <option value="40평">40평</option>  <option value="50평">50평</option>  </select>  </div>  <button onclick="save\_order()" type="button" class="btn btn-warning mybtn">주문하기</button>  </div>  <table class="table">  <thead>  <tr>  <th scope="col">이름</th>  <th scope="col">주소</th>  <th scope="col">평수</th>  </tr>  </thead>  <tbody id="order-box">  </tbody>  </table>  </div>  </body>  </html> |

서버측 App.py

| from flask import Flask, render\_template, request, jsonify  from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta  app = Flask(\_\_name\_\_)  @app.route('/')  def home():  return render\_template('index.html')  @app.route("/mars", methods=["POST"])  def web\_mars\_post():  # sample\_receive = request.form['sample\_give']  # print(sample\_receive)  # return jsonify({'msg': 'POST 연결 완료!'})  name\_receive = request.form['name\_give']  address\_receive = request.form['address\_give']  size\_receive = request.form['size\_give']  doc = {  'name':name\_receive,  'address':address\_receive,  'size':size\_receive  }  db.mars.insert\_one(doc)  return jsonify({'msg': '주문 완료!'})  @app.route("/mars", methods=["GET"])  def web\_mars\_get():  order\_list = list(db.mars.find({}, {'\_id':False}))  return jsonify({'orders': order\_list })  if \_\_name\_\_ == '\_\_main\_\_':  app.run('0.0.0.0', port=5000, debug=True) |
| --- |

결과 화면은

|  |
| --- |

[스파르타피디아]-프로젝트 세팅

1. **새 프로젝트 생성**

PyCharm구동>새 프로젝트>movie 폴더 지정 후 생성

**2. 기본 폴더/파일 생성**

templates, static 폴더 추가, app.py, index.html 파일 생성

**3. 패키지 설치**

Flask(웹서버), PyMongo, DNSPython(이상 DB), requests, bs4(웹 스크롤링) 5개 패키지 추가 (설치방법은 생략)

구성 완료!

**메타태크 크롤링 실습**

웹 크롤링 중 메타태그 크롤링에 대해 학습한다

일반적 웹 크롤링과는 조금 다른 방식이 이용된다.



페이지 검사를 수행 한 후 메타태그 부분을 조회, 제목(og:title), 이미지(og:image), 설명(og:description) 부분을 가져오게 된다

|  |
| --- |

meta\_prac.py 실습

| import requests  from bs4 import BeautifulSoup  url = 'https://movie.naver.com/movie/bi/mi/basic.naver?code=191597'  headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}  data = requests.get(url,headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  # 여기에 코딩을 해서 meta tag를 먼저 가져와보겠습니다.  # print(soup)  # title = soup.select\_one('meta[property="og:title"]')  # print(title)  # <meta content="보스 베이비 2" property="og:title"/>  title = soup.select\_one('meta[property="og:title"]')['content']  image = soup.select\_one('meta[property="og:image"]')['content']  desc = soup.select\_one('meta[property="og:description"]')['content']  print(title, image, desc) |
| --- |

결과를 확인하면 제대로 가져오는 것을 확인할 수 있다

| 보스 베이비 2 https://movie-phinf.pstatic.net/20210622\_174/1624324910624JhEq2\_JPEG/movie\_image.jpg?type=m665\_443\_2 베이비 주식회사의 레전드 보스 베이비에서 인생 만렙 CEO가 된 ‘테드’.베이비인 줄 알았던 조카 ‘티... |
| --- |

[스파르타피디아]-POST연습(포스팅하기)

서버측(app.py)

| import 영역 추가 |
| --- |
| # 웹 크롤링용 import  import requests  from bs4 import BeautifulSoup  # DB처리용 import  from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta |
| posting() 함수 처리 |
| def movie\_post():  # sample\_receive = request.form['sample\_give']  # print(sample\_receive)  # return jsonify({'msg': 'POST 연결 완료!'})  url\_receive = request.form['url\_give']  star\_receive = request.form['star\_give']  comment\_receive = request.form['comment\_give']  headers = {  'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}  data = requests.get(url\_receive, headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  title = soup.select\_one('meta[property="og:title"]')['content']  image = soup.select\_one('meta[property="og:image"]')['content']  desc = soup.select\_one('meta[property="og:description"]')['content']  doc = {  'title':title,  'image':image,  'desc':desc,  'star':star\_receive,  'comment':comment\_receive  }  db.movies.insert\_one(doc)  return jsonify({'msg': '저장 완료!'}) |

Client 처리(index.html)

| posting function 처리 |
| --- |
| function posting() {  let url = $('#url').val()  let star = $('#star').val()  let comment = $('#comment').val()  $.ajax({  type: 'POST',  url: '/movie',  data: {url\_give:url, star\_give:star, comment\_give:comment},  success: function (response) {  alert(response['msg'])  ***window***.location.reload()  }  }); |

입력 처리

|  |
| --- |

처리결과 DB 확인⇒ OK

|  |
| --- |

**[스파르타피디아]-GET 연습(보여주기)**

쌓인 데이터를 가져와서 보여주는 연습

서버 처리(app.py)

| movie\_get() 처리  DB에서 자료를 가져와서 리스트로 넘겨준다. 끝! |
| --- |
| def movie\_get():  movie\_list = list(db.movies.find({}, {'\_id':False}))  return jsonify({'movies':movie\_list}) |

Client 처리(index.html)

| listing() 함수 처리  1. 리스트를 가져와서 반복처리한다  2. 반복처리에서는   * 리스트 값을 처리한 후 (let comment = rows[i]['comment'] …) * temp\_html 문자열을 생성한 후 * cards-box에 추가 과정을 반복 |
| --- |
| function listing() {  $.ajax({  type: 'GET',  url: '/movie',  data: {},  success: function (response) {  // console.log(response['movies'])  let rows = response['movies']  for (let i = 0; i < rows.length; i++) {  let comment = rows[i]['comment']  let title = rows[i]['title']  let desc = rows[i]['desc']  let image = rows[i]['image']  let star = rows[i]['star']  let star\_image = '⭐'.repeat(star)  let temp\_html = `<div class="col">  <div class="card h-100">  <img src="${image}"  class="card-img-top">  <div class="card-body">  <h5 class="card-title">${title}</h5>  <p class="card-text">${desc}</p>  <p>${star\_image}</p>  <p class="mycomment">${comment}</p>  </div>  </div>  </div>`  $('#cards-box').append(temp\_html)  }  }  })  } |

결과를 보면 이렇게… 새로운 영화를 추가해도 OK.

|  |
| --- |

**숙제: 팬명록 등록/조회**

1주차 했던 팬명록 기록, 저장 및 조회기능 처리

1. **새 프로젝트 생성**

PyCharm구동>새 프로젝트>homework 폴더 지정 후 생성

**2. 기본 폴더/파일 생성**

templates, static 폴더 추가, app.py, index.html 파일 생성

**3. 패키지 설치**

Flask(웹서버), PyMongo, DNSPython(이상 DB), 3개 패키지 추가 (설치방법은 생략)

구성 완료!

서버측 처리(app.py)

| from flask import Flask, render\_template, request, jsonify  app = Flask(\_\_name\_\_)  # DB처리용 import  from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta  @app.route('/')  def home():  return render\_template('index.html')  @app.route("/homework", methods=["POST"])  def homework\_post():  # sample\_receive = request.form['sample\_give']  # print(sample\_receive)  # return jsonify({'msg': 'POST 연결 완료!'})  name = request.form['name\_give']  comment = request.form['comment\_give']  doc = {  'name':name,  'comment':comment  }  db.homework.insert\_one(doc)  return jsonify({'msg': '저장 완료!'})  @app.route("/homework", methods=["GET"])  def homework\_get():  # return jsonify({'msg': 'GET 연결 완료!'})  homework\_list = list(db.homework.find({}, {'\_id':False}))  return jsonify({'homework':homework\_list})  if \_\_name\_\_ == '\_\_main\_\_':  app.run('0.0.0.0', port=5000, debug=True) |
| --- |

Client측 처리(index.html)

| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  crossorigin="anonymous"></script>  <title>방탄소년단 팬명록</title>  <link href="https://fonts.googleapis.com/css2?family=Noto+Serif+KR:wght@200;300;400;500;600;700;900&display=swap"  rel="stylesheet">  <style>  \* {  font-family: 'Noto Serif KR', serif;  }  .mypic {  width: 100%;  height: 300px;  background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('http://images.khan.co.kr/article/2022/04/12/2022041202000648400119481.jpg');  background-position: center 30%;  background-size: cover;  color: white;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  }  .mypost {  width: 95%;  max-width: 500px;  margin: 20px auto 20px auto;  box-shadow: 0px 0px 3px 0px black;  padding: 20px;  }  .mypost > button {  margin-top: 15px;  }  .mycards {  width: 95%;  max-width: 500px;  margin: auto;  }  .mycards > .card {  margin-top: 10px;  margin-bottom: 10px;  }  </style>  <script>  $(***document***).ready(function () {  set\_temp()  show\_comment()  });  function set\_temp() {  $.ajax({  type: "GET",  url: "http://spartacodingclub.shop/sparta\_api/weather/seoul",  data: {},  success: function (response) {  $('#temp').text(response['temp'])  }  })  }  function save\_comment() {  let name = $('#name').val()  let comment = $('#comment').val()  $.ajax({  type: 'POST',  url: '/homework',  data: {name\_give:name, comment\_give:comment},  success: function (response) {  alert(response['msg'])  ***window***.location.reload()  }  })  }  function show\_comment() {  $.ajax({  type: "GET",  url: "/homework",  data: {},  success: function (response) {  let rows = response['homework']  for (let i = 0; i < rows.length; i++) {  let name = rows[i]['name']  let comment = rows[i]['comment']  let temp\_html = `<div class="card">  <div class="card-body">  <blockquote class="blockquote mb-0">  <p>${comment}</p>  <footer class="blockquote-footer">${name}</footer>  </blockquote>  </div>  </div>`  $('#comment-list').append(temp\_html)  }  }  })  }  </script>  </head>  <body>  <div class="mypic">  <h1>방탄소년단 팬명록</h1>  <p>현재기온: <span id="temp">36</span>도</p>  </div>  <div class="mypost">  <div class="form-floating mb-3">  <input type="text" class="form-control" id="name" placeholder="url">  <label for="floatingInput">닉네임</label>  </div>  <div class="form-floating">  <textarea class="form-control" placeholder="Leave a comment here" id="comment"  style="height: 100px"></textarea>  <label for="floatingTextarea2">응원댓글</label>  </div>  <button onclick="save\_comment()" type="button" class="btn btn-dark">응원 남기기</button>  </div>  <div class="mycards" id="comment-list">  </div>  </body>  </html> |
| --- |

**파이썬에서는 괄호 줄을 잘 맞추어야 한다.**

괄호 짝을 맞추느라 30분을 헤매었다.

이건 룰이니 지켜 주어야 할 것 같다.

**4주차 후기**

4 주차에는 Flask 프레임워크를 이용한 파이썬 웹 서버 프로그램을 주로 다루었다.

로컬 개발환경에서 웹서버와 클라이언트 프로그램을 개발하고, 원격지 MongoDB의 저장소를 활용하여 웹개발 환경을 구축하는 과정이었다.

화성땅 공동구매, 스파르타피디아, 팬명록 등 기존의 서비스들에 웹크롤링, 메타태그 크롤링, 원격 DB를 이용하여 자체적인 웹개발 프레임워크 기초 골격을 갖추어 나갈 수 있었다.

아직 5주차 수업은 남아 있지만, 4주간의 학습과정을 통하여 일반인들도 얼마든지 학습을 통해 코딩 전문가가 될 수 있을 정도로 개발방법이 쉬워졌음을 충분히 알고도 남음이 있다.

이제껏 개발자들이 누리던 특권이 사라지고 있음을 실감한다.

더군다나 낡은 구습에 얽매여 기존 개발 지식으로 개발자 역할을 하지 않고, 새로운 개발환경을 익히고자 했던 나의 선택이 얼마나다행이었는지 모르겠다.

마지막 5주차 학습을 남겨두고 있다.

완강 후에는 또 어떤 세상이 기다리고 있을지 자못 궁금해진다.