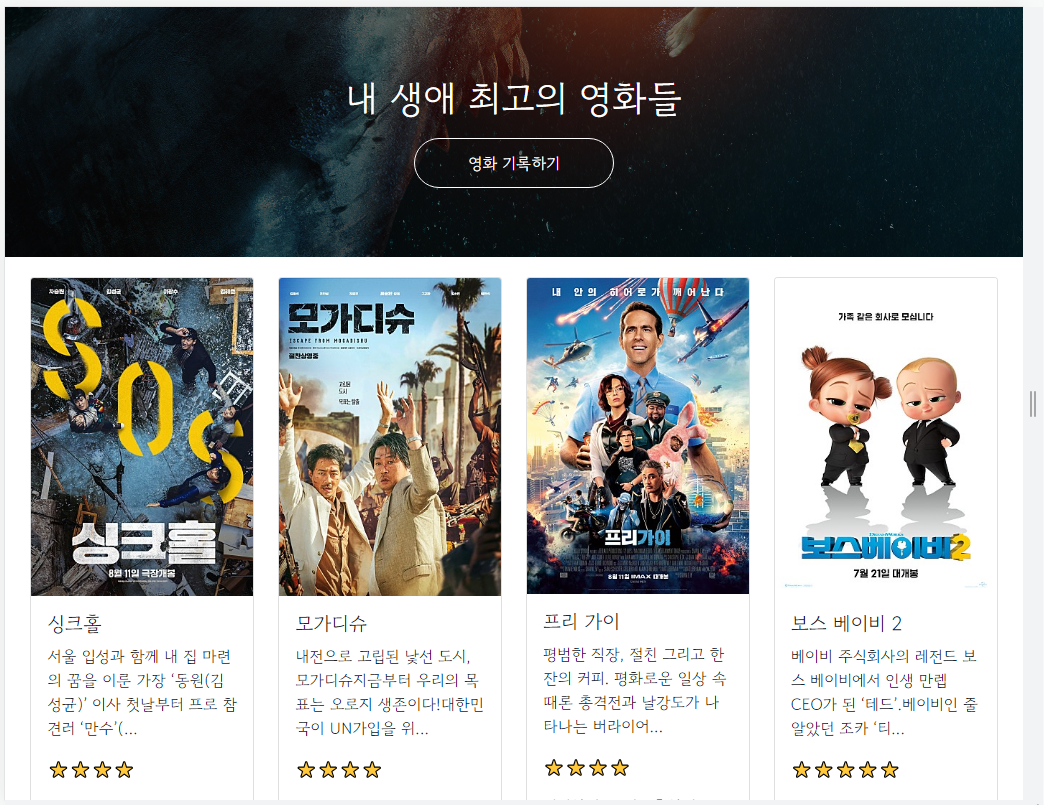
**웹개발 종합반 78기(튜터: 이범규)**

**수업노트 3주차 (22/05/06)**

1. 3주차 설치
2. 연습 겸 복습 - 스파르타피디앙[ OpenAPI 붙이기
3. 파이썬 시작하기
4. 파이썬 기초공부
5. 파이썬 패키지 설치하기
6. 패키지 사용해보기
7. 웹스크래핑(크롤링) 기초
8. Quiz\_웹 스크래핑(크롤링) 연습
9. DB개괄
10. mongoDB 시작하기
11. mongoDB 연결하기
12. pymongo로 DB조작하기
13. 웹 스크래핑결과 저장하기
14. Quiz\_웹 스크래핑결과 이용하기
15. 3주차 끝\_숙제 설명
16. 3주차 숙제 해설
17. 3주차 설치

- Python, gitbash 설치

1. 연습 겸 복습 - 스파르타피디아 OpenAPI 붙이기

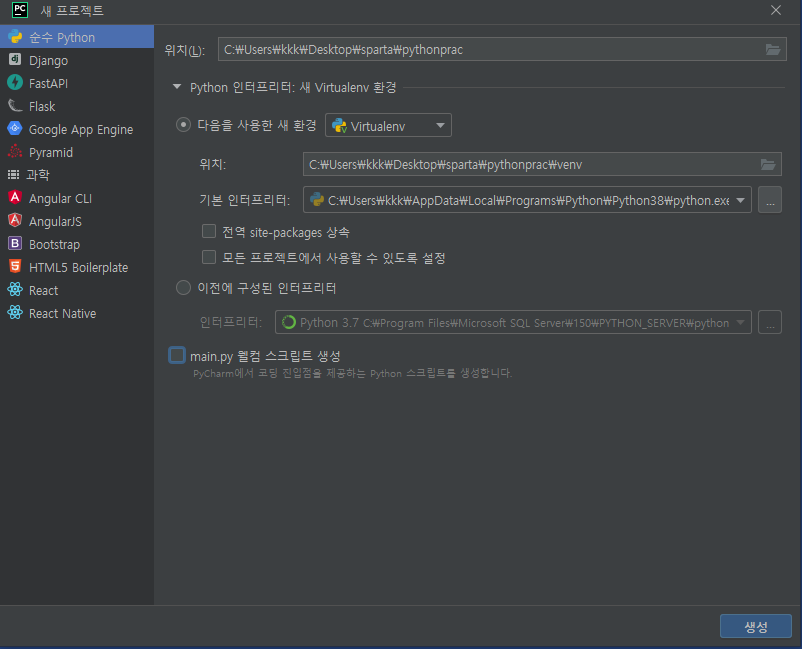


| <!doctype html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" 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>스파르타 피디아</title>  <link href="https://fonts.googleapis.com/css2?family=Gowun+Dodum&display=swap" rel="stylesheet">  <style>  \* {  font-family: 'Gowun Dodum', sans-serif;  }  .mytitle {  width: 100%;  height: 250px;  background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://movie-phinf.pstatic.net/20210715\_95/1626338192428gTnJl\_JPEG/movie\_image.jpg');  background-position: center;  background-size: cover;  color: white;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  }  .mytitle > button {  width: 200px;  height: 50px;  background-color: transparent;  color: white;  border-radius: 50px;  border: 1px solid white;  margin-top: 10px;  }  .mytitle > button:hover {  border: 2px solid white;  }  .mycomment {  color: gray;  }  .mycards {  margin: 20px auto 0px auto;  width: 95%;  max-width: 1200px;  }  .mypost {  width: 95%;  max-width: 500px;  margin: 20px auto 0px auto;  padding: 20px;  box-shadow: 0px 0px 3px 0px gray;  display: none;  }  .mybtns {  display: flex;  flex-direction: row;  align-items: center;  justify-content: center;  margin-top: 20px;  }  .mybtns > button {  margin-right: 10px;  }  </style>  <script>  $(***document***).ready(function(){  listing();  });  function listing() {  $('#cards-box').empty()  $.ajax({  type: "GET",  url: "http://spartacodingclub.shop/web/api/movie",  data: {},  success: function (response) {  let rows = response['movies']  for (let i = 0; i < rows.length; i++) {  let title = rows[i]['title']  let desc = rows[i]['desc']  let image = rows[i]['image']  let star = rows[i]['star']  let comment = rows[i]['comment']  let star\_image = '⭐'.repeat(star)  // console.log(title, desc, image, star, comment)  let temp\_html=`<div class="col">  <div class="card h-100">  <img src="${image}"  class="card-img-top" alt="...">  <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)  }  }  })  }  function open\_box(){  $('#post-box').show()  }  function close\_box(){  $('#post-box').hide()  }  </script>  </head>  <body>  <div class="mytitle">  <h1>내 생애 최고의 영화들</h1>  <button onclick="open\_box()">영화 기록하기</button>  </div>  <div class="mypost" id="post-box">  <div class="form-floating mb-3">  <input id="url" type="email" class="form-control" placeholder="name@example.com">  <label>영화URL</label>  </div>  <div class="input-group mb-3">  <label class="input-group-text" for="inputGroupSelect01">별점</label>  <select class="form-select" id="inputGroupSelect01">  <option selected>-- 선택하기 --</option>  <option value="1">⭐</option>  <option value="2">⭐⭐</option>  <option value="3">⭐⭐⭐</option>  <option value="4">⭐⭐⭐⭐</option>  <option value="5">⭐⭐⭐⭐⭐</option>  </select>  </div>  <div class="form-floating">  <textarea id="comment" class="form-control" placeholder="Leave a comment here" id="floatingTextarea2"  style="height: 100px"></textarea>  <label for="floatingTextarea2">코멘트</label>  </div>  <div class="mybtns">  <button type="button" class="btn btn-dark">기록하기</button>  <button onclick="close\_box()" type="button" class="btn btn-outline-dark">닫기</button>  </div>  </div>  <div class="mycards">  <div class="row row-cols-1 row-cols-md-4 g-4" id="cards-box">  <div class="col">  <div class="card h-100">  <img src="https://movie-phinf.pstatic.net/20210728\_221/1627440327667GyoYj\_JPEG/movie\_image.jpg"  class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">영화 제목이 들어갑니다</h5>  <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>  <p>⭐⭐⭐</p>  <p class="mycomment">나의 한줄 평을 씁니다</p>  </div>  </div>  </div>  <div class="col">  <div class="card h-100">  <img src="https://movie-phinf.pstatic.net/20210728\_221/1627440327667GyoYj\_JPEG/movie\_image.jpg"  class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">영화 제목이 들어갑니다</h5>  <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>  <p>⭐⭐⭐</p>  <p class="mycomment">나의 한줄 평을 씁니다</p>  </div>  </div>  </div>  <div class="col">  <div class="card h-100">  <img src="https://movie-phinf.pstatic.net/20210728\_221/1627440327667GyoYj\_JPEG/movie\_image.jpg"  class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">영화 제목이 들어갑니다</h5>  <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>  <p>⭐⭐⭐</p>  <p class="mycomment">나의 한줄 평을 씁니다</p>  </div>  </div>  </div>  </div>  </div>  </body>  </html> |
| --- |

1. 파이썬 시작하기

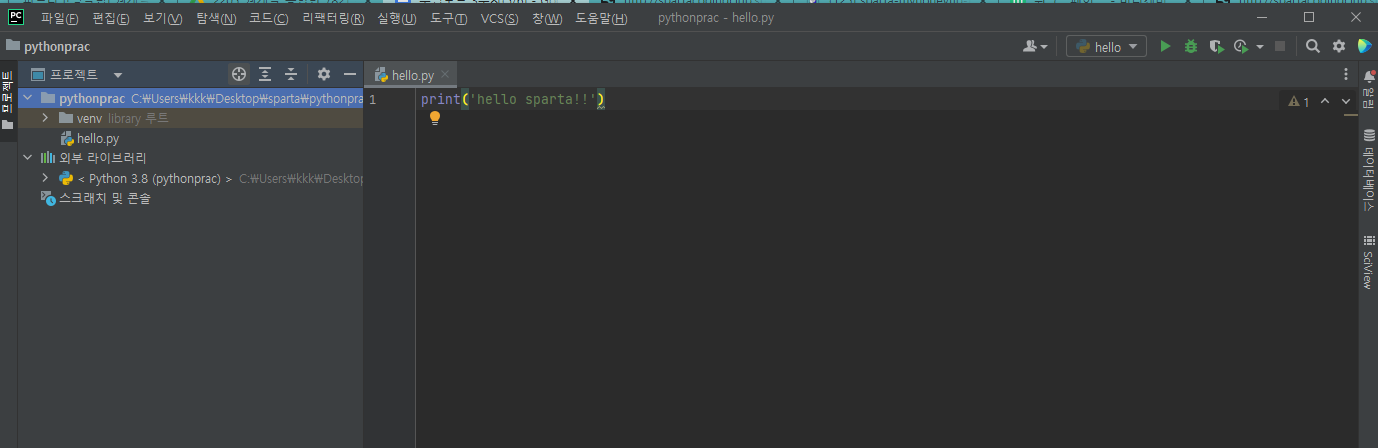
1. 프로젝트 폴더 생성(pythonprac)

**2. 새 프로젝트 추가(인터프리터 Python38 지정되어 있는지 확인!!!)**



설치 확인!

Hello!! 출력 확인



1. 파이쎤 기초공부

리스트, 딕셔너리, 함수, 조건문, 반복문(list), 반복문(dictionary) 문법 연습

| # 리스트  # a\_list = ['사과', '배', '감']  #  # a\_list.append('수박')  # print(a\_list)  #딕셔너리  # a\_dict = {  # 'name':'bob',  # 'age':27  # }  #  # print(a\_dict['name'])  #함수  # def sum(a, b):  # print('더하자!')  # return a+b  #  # result = sum(1, 2)  # print(result)  #조건문  # def is\_adult(age):  # if age > 20:  # print('성인입니다')  # else:  # print(' 청소년입니다')  #  # is\_adult(15)  #반복문(리스트)  # fruits = ['사과','배','배','감','수박','귤','딸기','사과','배','수박']  #  # # for fruit in fruits:  # # print(fruit)  #  # count=0  # for aaa in fruits:  # if aaa == '배':  # count += 1  # print(count)  #딕셔너리  people = [ {'name': 'bob', 'age': 20},  {'name': 'carry', 'age': 38},  {'name': 'john', 'age': 7},  {'name': 'smith', 'age': 17},  {'name': 'ben', 'age': 27}]  for ppp in people:  # print(ppp)  if ppp['age'] > 20:  print(ppp['name']) |
| --- |

1. 파이썬 패키지 설치하기

크롤링을 위한 패키지 설치가 필요함

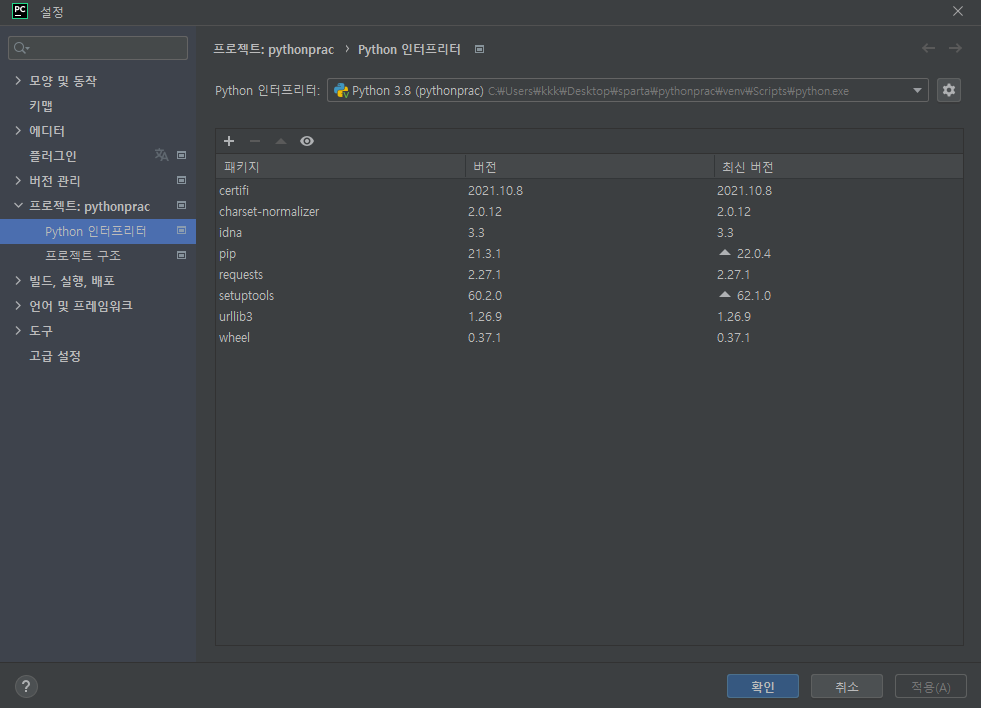
* 가상환경(virtual Environment: VENV)

: A사/B사 별로 파이썬 패키지 및 버전이 다를 경우, 프로젝트 환경을 서로 다르게 지정할 필요가 있다. 이 경우 가상환경을 쓴다.

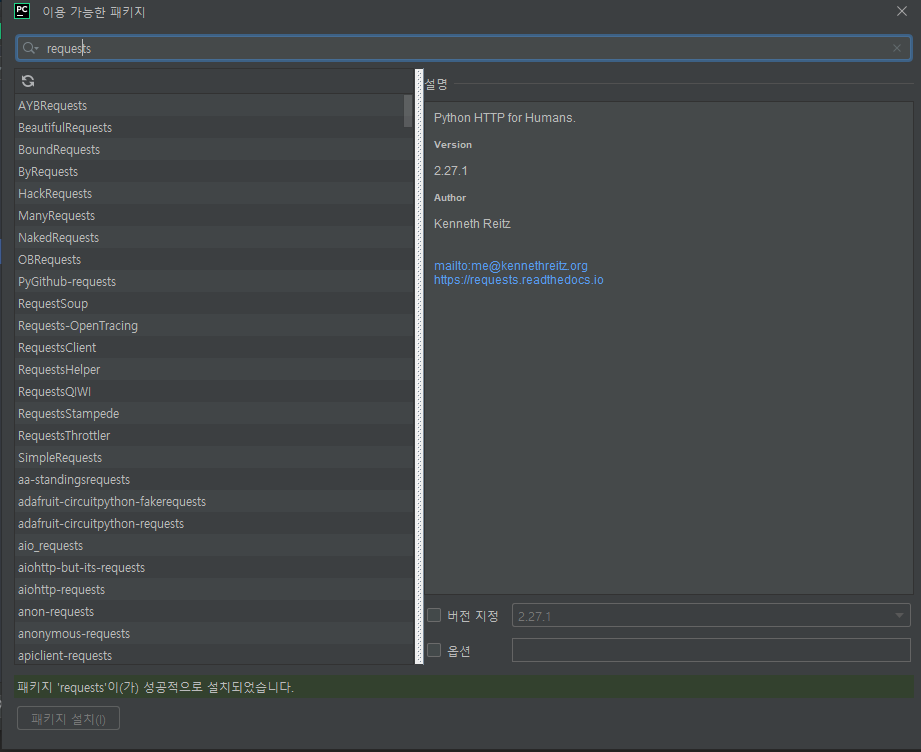


* **패키지 설치방법**

- 파일>설정>프로젝트:pythonprac>python인터프리터 (현재 설치된 패키지 목록 조회)



* + 선택 후 해당 패키지(requests) 선택 후 “패키지 설치” 선택 (패키지 설치됨)
* 프로젝트 패키지 목록에서 설치된 패키지(requests) 확인 가능함



1. 패키지 사용해보기

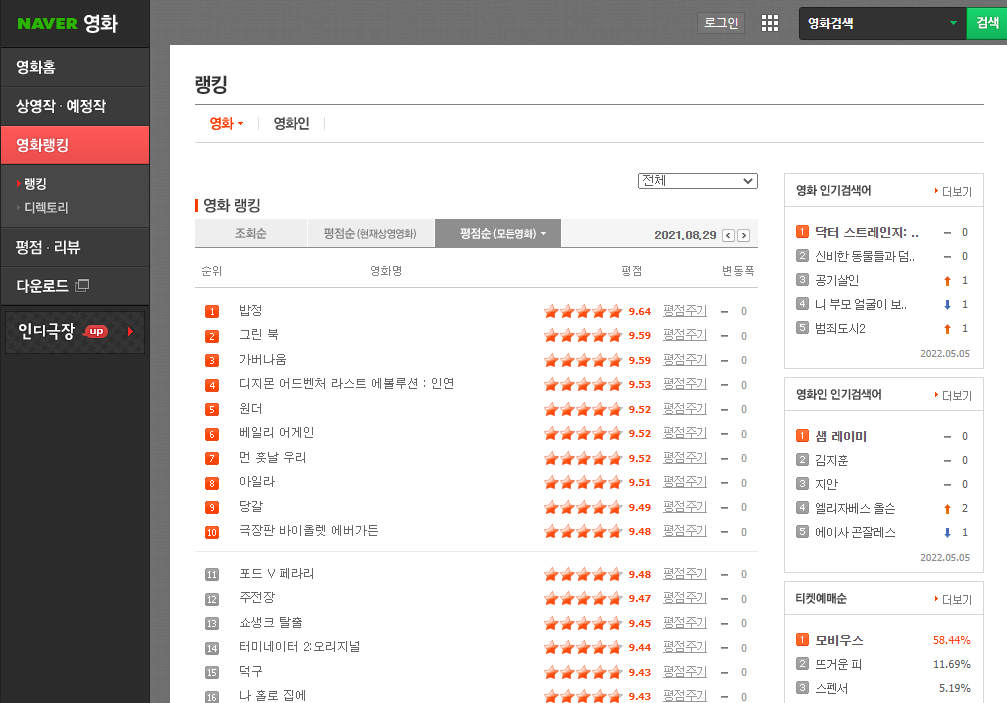
| import requests # requests 라이브러리 설치 필요  r = requests.get('http://spartacodingclub.shop/sparta\_api/seoulair')  rjson = r.json()  # Json 확인  # print(rjson)  rows = rjson['RealtimeCityAir']['row']  # print(rows)  for row in rows:  # print(row)  gu\_name = row['MSRSTE\_NM']  gu\_mise = row['IDEX\_MVL']  if gu\_mise < 60:  print(gu\_mise) |
| --- |

1. 웹스크래핑(크롤링) 기초

**크롤링용 패키지: Beautiful Soup (BS4)**

크롤링대상: 네이버영화

https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=pnt&date=20210829



\* 추가 패키지 설치(BS4)

1. 대상(화면의 밥정) 마우스 오른쪽>검사 선택 (html 소스 나타남)
2. **html 소스 선택 후 마우스 오른쪽 선택>Copy>Copy Selector 선택**

| import requests  from bs4 import BeautifulSoup  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('https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=pnt&date=20210829',headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  # print(soup)  #단일 레코드 처리(밥정 검사>Copy>Copy Selector)  # title = soup.select\_one('#old\_content > table > tbody > tr:nth-child(2) > td.title > div > a')  #  # print(title)  # print(title.text)  # print(title['href'])  #복수 레코드 처리 (그린 북, 가버나움 검사>Copy>Copy Selector)  #old\_content > table > tbody > tr:nth-child(3) > td.title > div > a  #old\_content > table > tbody > tr:nth-child(4) > td.title > div > a  #전체 레코드 가져오기  movies = soup.select('#old\_content > table > tbody > tr')  #출력해 보면 전체 레코드가 나타남  # print(movies)  #Parser 처리  for movie in movies:  # Parser 처리(td.title > div > a 하단 태그 목록을 가져옴) 단. None(가로선) 레코드도 같이 가져온다  a = movie.select\_one('td.title > div > a')  #print(a)  # None을 제외하고 프린트하면  if a is not None:  #print(a)  # 영화 제목만을 가져오려면  print(a.text) |
| --- |

1. Quiz\_웹 스크래핑(크롤링) 연습

이쁘게 가져오기

| import requests  from bs4 import BeautifulSoup  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('https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=pnt&date=20210829',headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  # print(soup)  #단일 레코드 처리(밥정 검사>Copy>Copy Selector)  # title = soup.select\_one('#old\_content > table > tbody > tr:nth-child(2) > td.title > div > a')  #  # print(title)  # print(title.text)  # print(title['href'])  #복수 레코드 처리 (그린 북, 가버나움 검사>Copy>Copy Selector)  #old\_content > table > tbody > tr:nth-child(3) > td.title > div > a  #old\_content > table > tbody > tr:nth-child(4) > td.title > div > a  #전체 레코드 가져오기  movies = soup.select('#old\_content > table > tbody > tr')  #출력해 보면 전체 레코드가 나타남  #print(movies)  #Parser 처리  for movie in movies:  # Parser 처리(td.title > div > a 하단 태그 목록을 가져옴) 단. None(가로선) 레코드도 같이 가져온다  a = movie.select\_one('td.title > div > a')  # None을 제외하고 프린트하면  if a is not None:  #print(a)  # 영화 제목만을 가져오려면  title = a.text  # 순위 가져오기  # old\_content > table > tbody > tr:nth-child(2) > td:nth-child(1) > img  rank = movie.select\_one('td:nth-child(1) > img')['alt']  # 평점 가져오기  # old\_content > table > tbody > tr:nth-child(3) > td.point  star = movie.select\_one('td.point').text  print(rank, title, star)  #순위, 평점 가져오기 |
| --- |

결과는 순위, 영화제목, 평점 순으로 이쁘게 가져온다

| 01 밥정 9.64  02 그린 북 9.59  03 가버나움 9.59  04 디지몬 어드벤처 라스트 에볼루션 : 인연 9.53 |
| --- |

1. DB개괄

SQL vs NOSQL(Not Only SQL)

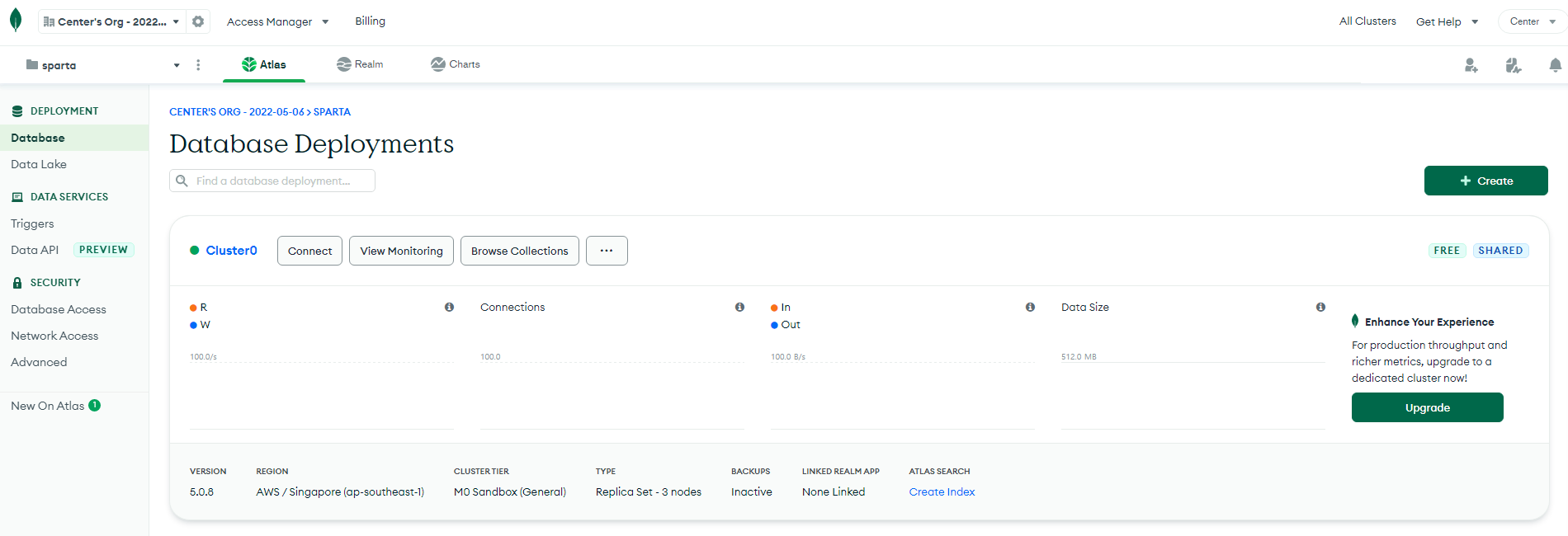
1. mongoDB 시작하기

mongoDB 설치하기

<https://account.mongodb.com/account/register>

mongoDB 회원가입([99damoi@gmail.com](mailto:99damoi@gmail.com))

id, pwd: test/sparta 설정



1. mongoDB 연결하기

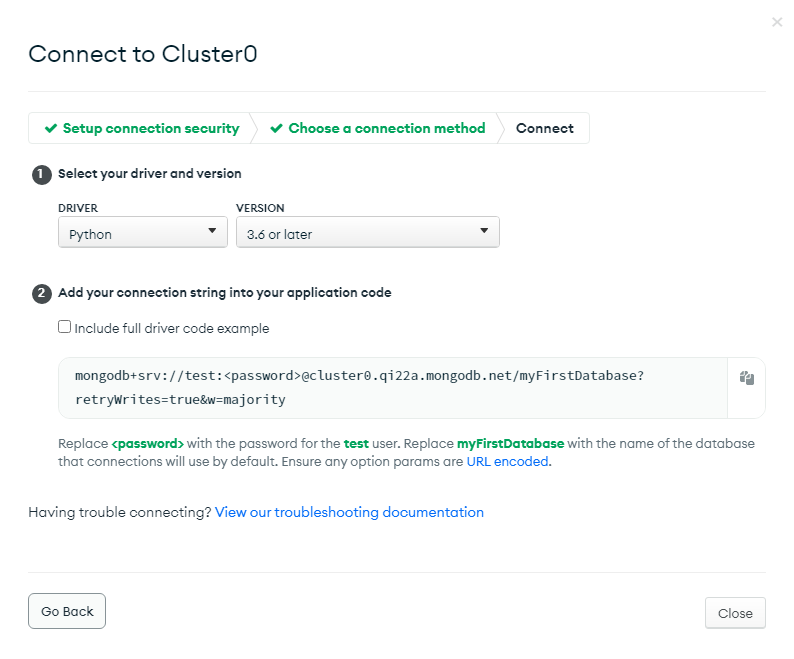
**1. 패키지 추가 설치**

**python에서 mongoDB를 연결하기 위해서는 pymongo, dnspython 패키지가 필요함→ 설치**

**2. mongoDB 연결**

**위 mongoDB화면에서 “Connect>Connect Your Application 선택(아래화면 나타남)**

* **Driver: python, version: 3.6 or later 선택**

****

**3. PyCham에서 새 파일 생성(dbprac.py)**

**코드 입력**

| **from pymongo import MongoClient**  **client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')**  **db = client.dbsparta** |
| --- |

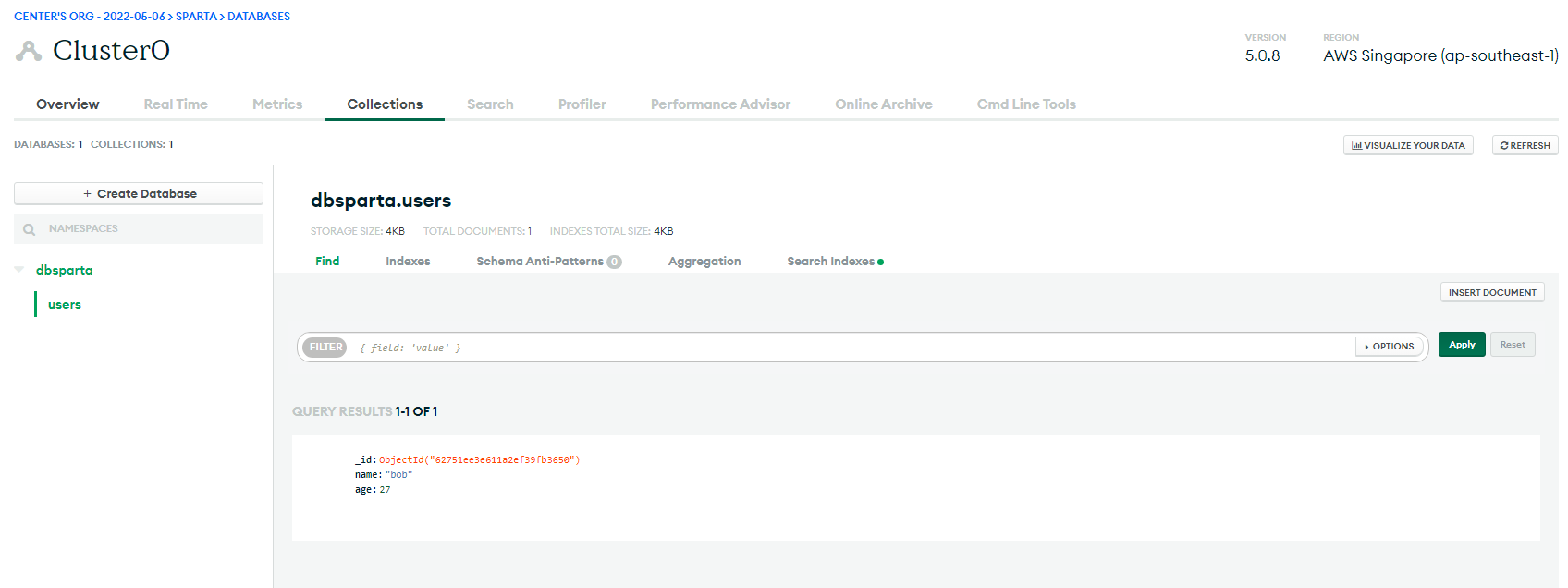
**\* MongoClient 접속 문자열에 위 화면의 접속문자열 복붙 후**

**password(sparta), DB명(myFirstDB -> cluster0) 변경 후 저장**

**DB 레코드 처리 테스트하기**

| **from pymongo import MongoClient**  **client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')**  **db = client.dbsparta**  **doc = {**  **'name':'bob',**  **'age':27**  **}**  **db.users.insert\_one(doc)** |
| --- |

**mongoDB 저장결과 확인**



mongoDB내 users 테이블에 방금 입력한 데이터가 저장되어 있는 것을 확인할 수 있다! -> 성공

1. pymongo로 DB조작하기

insert 테스트

| db.users.insert\_one({'name':'bobby','age':27})  db.users.insert\_one({'name':'john','age':20})  db.users.insert\_one({'name':'ann','age':28}) |
| --- |
|  |

select 테스트

| # select 조회 처리  all\_users = list(db.users.find({}))  for user in all\_users:  print(user) |
| --- |
| C:\Users\kkk\Desktop\sparta\pythonprac\venv\Scripts\python.exe C:/Users/kkk/Desktop/sparta/pythonprac/dbprac.py  {'\_id': ObjectId('62751ee3e611a2ef39fb3650'), 'name': 'bob', 'age': 27}  {'\_id': ObjectId('6275206409895b0e227ed742'), 'name': 'bobby', 'age': 27}  {'\_id': ObjectId('6275206509895b0e227ed743'), 'name': 'john', 'age': 20}  {'\_id': ObjectId('6275206509895b0e227ed744'), 'name': 'ann', 'age': 28} |

보기 어지러우니까

| # select 조회 처리  all\_users = list(db.users.find({}, {'\_id':False}))  for user in all\_users:  print(user) |
| --- |
| {'name': 'bob', 'age': 27}  {'name': 'bobby', 'age': 27}  {'name': 'john', 'age': 20}  {'name': 'ann', 'age': 28} |

한 레코드만 찾을 때는

| # 특정 레코드 검색 처리  user = db.users.find\_one({'name':'bobby'})  print(user) |
| --- |
| {'\_id': ObjectId('6275206409895b0e227ed742'), 'name': 'bobby', 'age': 27} |

보기 어지러우니

| # 특정 레코드 검색 처리  user = db.users.find\_one({'name':'bobby'}, {'\_id':False})  print(user) |
| --- |
| {'name': 'bobby', 'age': 27} |

나이만 가져오고 싶을 땐

| # 특정 레코드 검색 처리  user = db.users.find\_one({'name':'bobby'}, {'\_id':False})  print(user['age']) |
| --- |
| 27 |

레코드 수정 및 결과확인

| # 레코드 수정  db.users.update\_one({'name':'bobby'},{'$set':{'age':19}})  #수정결과 확인  user = db.users.find\_one({'name':'bobby'}, {'\_id':False})  print(user['age']) |
| --- |
| 19 |

레코드 삭제 및 결과 확인

| #레코드 삭제  db.users.delete\_one({'name':'bobby'})  #삭제결과 확인  all\_users = list(db.users.find({}, {'\_id':False}))  for user in all\_users:  print(user) |
| --- |
| {'name': 'bob', 'age': 27}  {'name': 'john', 'age': 20}  {'name': 'ann', 'age': 28} |

1. 웹 스크래핑결과 저장하기

영화 스크롤링 결과를 movies 테이블에 입력하기

| import requests  from bs4 import BeautifulSoup  from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta  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('https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=pnt&date=20210829',headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  #전체 레코드 가져오기  movies = soup.select('#old\_content > table > tbody > tr')  #Parser 처리  for movie in movies:  # Parser 처리(td.title > div > a 하단 태그 목록을 가져옴) 단. None(가로선) 레코드도 같이 가져온다  a = movie.select\_one('td.title > div > a')  # None을 제외하고 프린트하면  if a is not None:  #print(a)  # 영화 제목만을 가져오려면  title = a.text  # 순위 가져오기  # old\_content > table > tbody > tr:nth-child(2) > td:nth-child(1) > img  rank = movie.select\_one('td:nth-child(1) > img')['alt']  # 평점 가져오기  # old\_content > table > tbody > tr:nth-child(3) > td.point  star = movie.select\_one('td.point').text  # print(rank, title, star)  # DB 처리용 Dictionary 생성  doc = {  'title':title,  'rank':rank,  'star':star  }  # movies 테이블에 입력  db.movies.insert\_one(doc) |
| --- |
|  |

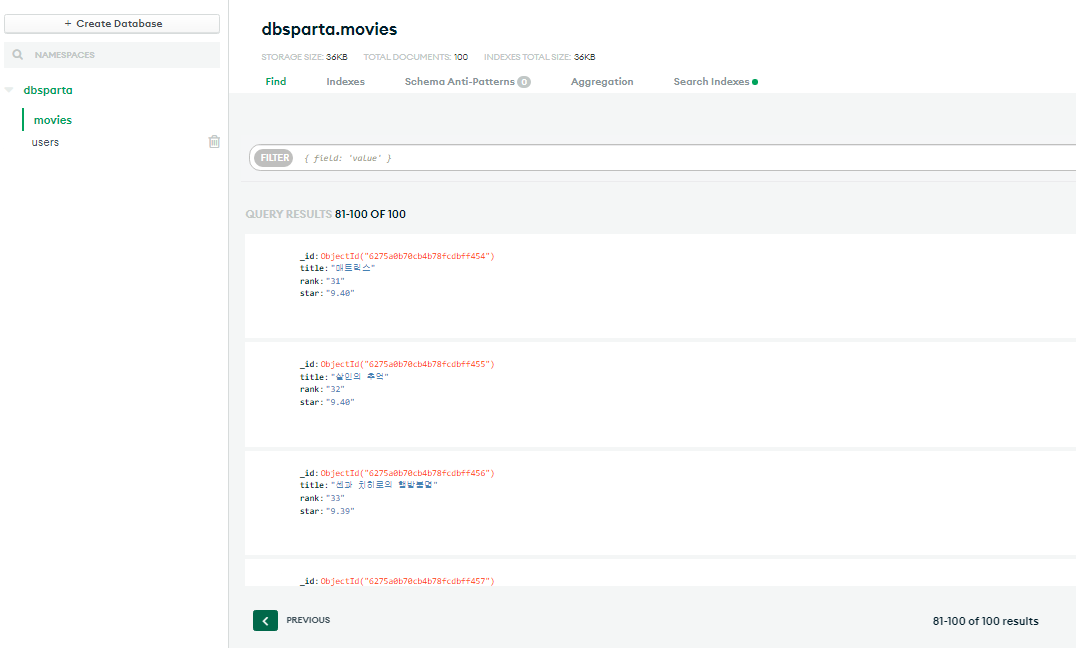
1. Quiz\_웹 스크래핑결과 이용하기

웹 스크래핑 결과를 생성한 DB에 저장해 본다

1. 웹 스크래핑 결과를 DB에 Insert

| import requests  from bs4 import BeautifulSoup  from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta  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('https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=pnt&date=20210829',headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  #전체 레코드 가져오기  movies = soup.select('#old\_content > table > tbody > tr')  #Parser 처리  for movie in movies:  # Parser 처리(td.title > div > a 하단 태그 목록을 가져옴) 단. None(가로선) 레코드도 같이 가져온다  a = movie.select\_one('td.title > div > a')  # None을 제외하고 프린트하면  if a is not None:  #print(a)  # 영화 제목만을 가져오려면  title = a.text  # 순위 가져오기  # old\_content > table > tbody > tr:nth-child(2) > td:nth-child(1) > img  rank = movie.select\_one('td:nth-child(1) > img')['alt']  # 평점 가져오기  # old\_content > table > tbody > tr:nth-child(3) > td.point  star = movie.select\_one('td.point').text  # print(rank, title, star)  # DB 처리용 Dictionary 생성  doc = {  'title':title,  'rank':rank,  'star':star  }  # movies 테이블에 입력  db.movies.insert\_one(doc) |
| --- |

2. DB 결과 확인



프로그램 2번 돌렸더니 곱배기로 저장되었다

다시, DB 자료를 Select해 오기

| from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta  # movies 전체 가져오기  all\_movies = list(db.movies.find({}, {'\_id':False}))  for movie in all\_movies:  print(movie)  # {'title': '죽은 시인의 사회', 'rank': '41', 'star': '9.38'} |
| --- |

결과조회해 보니 딱 2배. 성공!

| {'title': '밥정', 'rank': '01', 'star': '9.64'}  {'title': '그린 북', 'rank': '02', 'star': '9.59'}  {'title': '가버나움', 'rank': '03', 'star': '0'}  {'title': '디지몬 어드벤처 라스트 에볼루션 : 인연', 'rank': '04', 'star': '9.53'}  {'title': '원더', 'rank': '05', 'star': '9.52'}  {'title': '베일리 어게인', 'rank': '06', 'star': '9.52'}  {'title': '먼 훗날 우리', 'rank': '07', 'star': '9.52'}  {'title': '아일라', 'rank': '08', 'star': '9.51'}  {'title': '당갈', 'rank': '09', 'star': '9.49'}  {'title': '극장판 바이올렛 에버가든', 'rank': '010', 'star': '9.48'}  {'title': '포드 V 페라리', 'rank': '11', 'star': '9.48'}  {'title': '주전장', 'rank': '12', 'star': '9.47'}  {'title': '쇼생크 탈출', 'rank': '13', 'star': '9.45'}  {'title': '터미네이터 2:오리지널', 'rank': '14', 'star': '9.44'}  {'title': '덕구', 'rank': '15', 'star': '9.43'}  {'title': '나 홀로 집에', 'rank': '16', 'star': '9.43'}  {'title': '라이언 일병 구하기', 'rank': '17', 'star': '9.43'}  {'title': '클래식', 'rank': '18', 'star': '9.42'}  {'title': '잭 스나이더의 저스티스 리그', 'rank': '19', 'star': '9.42'}  {'title': '그대, 고맙소 : 김호중 생애 첫 팬미팅 무비', 'rank': '20', 'star': '9.42'}  {'title': '월-E', 'rank': '21', 'star': '9.42'}  {'title': '보헤미안 랩소디', 'rank': '22', 'star': '9.42'}  {'title': '사운드 오브 뮤직', 'rank': '23', 'star': '9.41'}  {'title': '포레스트 검프', 'rank': '24', 'star': '9.41'}  {'title': '빽 투 더 퓨쳐', 'rank': '25', 'star': '9.41'}  {'title': '위대한 쇼맨', 'rank': '26', 'star': '9.41'}  {'title': '글래디에이터', 'rank': '27', 'star': '9.41'}  {'title': '헬프', 'rank': '28', 'star': '9.41'}  {'title': '인생은 아름다워', 'rank': '29', 'star': '9.40'}  {'title': '타이타닉', 'rank': '30', 'star': '9.40'}  {'title': '매트릭스', 'rank': '31', 'star': '9.40'}  {'title': '살인의 추억', 'rank': '32', 'star': '9.40'}  {'title': '센과 치히로의 행방불명', 'rank': '33', 'star': '9.39'}  {'title': '토이 스토리 3', 'rank': '34', 'star': '9.39'}  {'title': '가나의 혼인잔치: 언약', 'rank': '35', 'star': '9.39'}  {'title': '헌터 킬러', 'rank': '36', 'star': '9.39'}  {'title': '캐스트 어웨이', 'rank': '37', 'star': '9.38'}  {'title': '집으로...', 'rank': '38', 'star': '9.38'}  {'title': '아이즈 온 미 : 더 무비', 'rank': '39', 'star': '9.38'}  {'title': '반지의 제왕: 왕의 귀환', 'rank': '40', 'star': '9.38'}  {'title': '죽은 시인의 사회', 'rank': '41', 'star': '9.38'}  {'title': '히든 피겨스', 'rank': '42', 'star': '9.38'}  {'title': '알라딘', 'rank': '43', 'star': '9.38'}  {'title': '어벤져스: 엔드게임', 'rank': '44', 'star': '9.38'}  {'title': '레옹', 'rank': '45', 'star': '9.38'}  {'title': '쉰들러 리스트', 'rank': '46', 'star': '9.37'}  {'title': '아이 캔 스피크', 'rank': '47', 'star': '9.37'}  {'title': '동주', 'rank': '48', 'star': '9.37'}  {'title': '안녕 베일리', 'rank': '49', 'star': '9.37'}  {'title': '클레멘타인', 'rank': '50', 'star': '9.37'}  {'title': '밥정', 'rank': '01', 'star': '9.64'}  {'title': '그린 북', 'rank': '02', 'star': '9.59'}  {'title': '가버나움', 'rank': '03', 'star': '9.59'}  {'title': '디지몬 어드벤처 라스트 에볼루션 : 인연', 'rank': '04', 'star': '9.53'}  {'title': '원더', 'rank': '05', 'star': '9.52'}  {'title': '베일리 어게인', 'rank': '06', 'star': '9.52'}  {'title': '먼 훗날 우리', 'rank': '07', 'star': '9.52'}  {'title': '아일라', 'rank': '08', 'star': '9.51'}  {'title': '당갈', 'rank': '09', 'star': '9.49'}  {'title': '극장판 바이올렛 에버가든', 'rank': '010', 'star': '9.48'}  {'title': '포드 V 페라리', 'rank': '11', 'star': '9.48'}  {'title': '주전장', 'rank': '12', 'star': '9.47'}  {'title': '쇼생크 탈출', 'rank': '13', 'star': '9.45'}  {'title': '터미네이터 2:오리지널', 'rank': '14', 'star': '9.44'}  {'title': '덕구', 'rank': '15', 'star': '9.43'}  {'title': '나 홀로 집에', 'rank': '16', 'star': '9.43'}  {'title': '라이언 일병 구하기', 'rank': '17', 'star': '9.43'}  {'title': '클래식', 'rank': '18', 'star': '9.42'}  {'title': '잭 스나이더의 저스티스 리그', 'rank': '19', 'star': '9.42'}  {'title': '그대, 고맙소 : 김호중 생애 첫 팬미팅 무비', 'rank': '20', 'star': '9.42'}  {'title': '월-E', 'rank': '21', 'star': '9.42'}  {'title': '보헤미안 랩소디', 'rank': '22', 'star': '9.42'}  {'title': '사운드 오브 뮤직', 'rank': '23', 'star': '9.41'}  {'title': '포레스트 검프', 'rank': '24', 'star': '9.41'}  {'title': '빽 투 더 퓨쳐', 'rank': '25', 'star': '9.41'}  {'title': '위대한 쇼맨', 'rank': '26', 'star': '9.41'}  {'title': '글래디에이터', 'rank': '27', 'star': '9.41'}  {'title': '헬프', 'rank': '28', 'star': '9.41'}  {'title': '인생은 아름다워', 'rank': '29', 'star': '9.40'}  {'title': '타이타닉', 'rank': '30', 'star': '9.40'}  {'title': '매트릭스', 'rank': '31', 'star': '9.40'}  {'title': '살인의 추억', 'rank': '32', 'star': '9.40'}  {'title': '센과 치히로의 행방불명', 'rank': '33', 'star': '9.39'}  {'title': '토이 스토리 3', 'rank': '34', 'star': '9.39'}  {'title': '가나의 혼인잔치: 언약', 'rank': '35', 'star': '9.39'}  {'title': '헌터 킬러', 'rank': '36', 'star': '9.39'}  {'title': '캐스트 어웨이', 'rank': '37', 'star': '9.38'}  {'title': '집으로...', 'rank': '38', 'star': '9.38'}  {'title': '아이즈 온 미 : 더 무비', 'rank': '39', 'star': '9.38'}  {'title': '반지의 제왕: 왕의 귀환', 'rank': '40', 'star': '9.38'}  {'title': '죽은 시인의 사회', 'rank': '41', 'star': '9.38'}  {'title': '히든 피겨스', 'rank': '42', 'star': '9.38'}  {'title': '알라딘', 'rank': '43', 'star': '9.38'}  {'title': '어벤져스: 엔드게임', 'rank': '44', 'star': '9.38'}  {'title': '레옹', 'rank': '45', 'star': '9.38'}  {'title': '쉰들러 리스트', 'rank': '46', 'star': '9.37'}  {'title': '아이 캔 스피크', 'rank': '47', 'star': '9.37'}  {'title': '동주', 'rank': '48', 'star': '9.37'}  {'title': '안녕 베일리', 'rank': '49', 'star': '9.37'}  {'title': '클레멘타인', 'rank': '50', 'star': '9.37'} |
| --- |

1. 3주차 끝\_숙제 설명

지니뮤직 사이트에서 순위, 제목, 가수명 가져오기 스크롤링하기

|  |
| --- |

결과는 이렇게 나와야 한다

| 1 MSG워너비 (M.O.M) 바라만 본다  2 aespa Next Level  3 이무진 신호등 |
| --- |

착수

| import requests  from bs4 import BeautifulSoup  from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta  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('https://www.genie.co.kr/chart/top200?ditc=M&rtm=N&ymd=20210701',headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  # print(soup)  #순위  #body-content > div.newest-list > div > table > tbody > tr:nth-child(1) > td.number  #body-content > div.newest-list > div > table > tbody > tr:nth-child(2) > td.number  # 제목  # body-content > div.newest-list > div > table > tbody > tr:nth-child(1) > td.info > a.title.ellipsis  #body-content > div.newest-list > div > table > tbody > tr:nth-child(2) > td.info > a.title.ellipsis  # 가수  #body-content > div.newest-list > div > table > tbody > tr:nth-child(1) > td.info > a.artist.ellipsis  #body-content > div.newest-list > div > table > tbody > tr:nth-child(2) > td.info > a.artist.ellipsis  #전체 레코드 가져오기  genies = soup.select('#body-content > div.newest-list > div > table > tbody > tr')  # print(genies)  #Parser 처리  for genie in genies:  # Parser 처리(td.title > div > a 하단 태그 목록을 가져옴) 단. None(가로선) 레코드도 같이 가져온다  rank = genie.select\_one('td.number').text[0:2].strip()  title = genie.select\_one('a.title.ellipsis').text.strip()  singer = genie.select\_one('a.artist.ellipsis').text.strip()  print(rank, singer, title) |
| --- |

결과를 보니

| 1 MSG워너비 (M.O.M) 바라만 본다  2 aespa Next Level  3 이무진 신호등  4 태연 (TAEYEON) Weekend  5 브레이브걸스 (Brave girls) 치맛바람 (Chi Mat Ba Ram)  6 방탄소년단 Butter  7 MSG워너비 (정상동기) 나를 아는 사람  8 방탄소년단 Permission to Dance  9 에픽하이 (EPIK HIGH) 비 오는 날 듣기 좋은 노래 (Feat. Colde)  10 헤이즈 (Heize) 헤픈 우연  11 빅마마 (Big Mama) 하루만 더  12 이무진 비와 당신  13 TWICE (트와이스) Alcohol-Free  14 브레이브걸스 (Brave girls) 롤린 (Rollin')  15 Justin Bieber 19금          Peaches (Feat. Daniel Caesar & Giveon)  16 오마이걸 (OH MY GIRL) Dun Dun Dance  17 방탄소년단 Dynamite  18 아이유 (IU) 라일락  19 조이 (JOY) 안녕 (Hello)  20 장범준 추적이는 여름 비가 되어 |
| --- |

잘 나가다가 한 놈이 말썽이다. 쓸데 없는 놈이 들어갔다.

다시 구글링 착수!

re.sub 함수 발견!

다시 도전!

| import requests  import re  from bs4 import BeautifulSoup  from pymongo import MongoClient  client = MongoClient('mongodb+srv://test:sparta@cluster0.qi22a.mongodb.net/Cluster0?retryWrites=true&w=majority')  db = client.dbsparta  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('https://www.genie.co.kr/chart/top200?ditc=M&rtm=N&ymd=20210701',headers=headers)  soup = BeautifulSoup(data.text, 'html.parser')  # print(soup)  #순위  #body-content > div.newest-list > div > table > tbody > tr:nth-child(1) > td.number  #body-content > div.newest-list > div > table > tbody > tr:nth-child(2) > td.number  # 제목  # body-content > div.newest-list > div > table > tbody > tr:nth-child(1) > td.info > a.title.ellipsis  #body-content > div.newest-list > div > table > tbody > tr:nth-child(2) > td.info > a.title.ellipsis  # 가수  #body-content > div.newest-list > div > table > tbody > tr:nth-child(1) > td.info > a.artist.ellipsis  #body-content > div.newest-list > div > table > tbody > tr:nth-child(2) > td.info > a.artist.ellipsis  #전체 레코드 가져오기  genies = soup.select('#body-content > div.newest-list > div > table > tbody > tr')  # print(genies)  #Parser 처리  for genie in genies:  # Parser 처리(td.title > div > a 하단 태그 목록을 가져옴) 단. None(가로선) 레코드도 같이 가져온다  rank = genie.select\_one('td.number').text[0:2].strip()  title = str(genie.select\_one('a.title.ellipsis').text.strip())  title = re.sub('19금', '', title, 0).strip()  singer = genie.select\_one('a.artist.ellipsis').text.strip()  print(rank, singer, title) |
| --- |

결과 확인해 보니 깔끔하다! 숙제 성공!

| 1 MSG워너비 (M.O.M) 바라만 본다  2 aespa Next Level  3 이무진 신호등  4 태연 (TAEYEON) Weekend  5 브레이브걸스 (Brave girls) 치맛바람 (Chi Mat Ba Ram)  6 방탄소년단 Butter  7 MSG워너비 (정상동기) 나를 아는 사람  8 방탄소년단 Permission to Dance  9 에픽하이 (EPIK HIGH) 비 오는 날 듣기 좋은 노래 (Feat. Colde)  10 헤이즈 (Heize) 헤픈 우연  11 빅마마 (Big Mama) 하루만 더  12 이무진 비와 당신  13 TWICE (트와이스) Alcohol-Free  14 브레이브걸스 (Brave girls) 롤린 (Rollin')  15 Justin Bieber Peaches (Feat. Daniel Caesar & Giveon)  16 오마이걸 (OH MY GIRL) Dun Dun Dance  17 방탄소년단 Dynamite  18 아이유 (IU) 라일락  19 조이 (JOY) 안녕 (Hello)  20 장범준 추적이는 여름 비가 되어  21 브레이브걸스 (Brave girls) 운전만해 (We Ride)  22 아이유 (IU) Celebrity  23 로꼬 & 이성경 러브 (Prod. by 로코베리)  24 Ed Sheeran Bad Habits  25 MSG워너비 상상더하기  26 STAYC (스테이씨) ASAP  27 라붐 (LABOUM) 상상더하기  28 원슈타인 밤이 되니까  29 SG워너비 Timeless  30 조정석 좋아좋아  31 Jawsh 685 & Jason Derulo & 방탄소년단 Savage Love (Laxed - Siren Beat) (BTS Remix)  32 경서예지 & 전건호 다정히 내 이름을 부르면  33 아이유 (IU) 내 손을 잡아  34 호미들 사이렌 Remix (Feat. UNEDUCATED KID & Paul Blanco)  35 Pink Sweat$ At My Worst  36 방탄소년단 작은 것들을 위한 시 (Boy With Luv) (Feat. Halsey)  37 디핵 (D-Hack) & PATEKO OHAYO MY NIGHT  38 김대명 가을 우체국 앞에서  39 장범준 나는 너 좋아  40 ASH ISLAND 멜로디  41 아이유 (IU) Blueming  42 송이한 밝게 빛나는 별이 되어 비춰줄게  43 아이유 (IU) 에잇 (Prod. & Feat. SUGA of BTS)  44 Anne-Marie 2002  45 양요섭 & 정은지 LOVE DAY (2021) (바른연애 길잡이 X 양요섭, 정은지)  46 조정석 아로하  47 장범준 흔들리는 꽃들 속에서 네 샴푸향이 느껴진거야  48 임영웅 이제 나만 믿어요  49 AKMU (악뮤) 낙하 (With 아이유)  50 Justin Bieber Off My Face |
| --- |

3주차 후기

1. 3주차 설치
2. 연습 겸 복습 - 스파르타피디앙[ OpenAPI 붙이기
3. 파이썬 시작하기
4. 파이썬 기초공부
5. 파이썬 패키지 설치하기
6. 패키지 사용해보기
7. 웹스크래핑(크롤링) 기초
8. Quiz\_웹 스크래핑(크롤링) 연습
9. DB개괄
10. mongoDB 시작하기
11. mongoDB 연결하기
12. pymongo로 DB조작하기
13. 웹 스크래핑결과 저장하기
14. Quiz\_웹 스크래핑결과 이용하기
15. 3주차 끝\_숙제 설명
16. 3주차 숙제 해설

3주차엔 서버 프로그래밍 중 파이썬과 DB 처리방법을 다루었다.

파이썬 프로그램을 설치하고,

웹 스크롤링 방법을 배우고,

DB를 설치하고,

웹 스크롤링 결과를 DB에 입력하고

DB에 저장된 값들을 조회, 수정, 삭제하는 법을 배웠다.

다시 한번 1주차, 2주차 배운 내용을 들여다 본다.

1주차에 HTML, CSS, Javascript를 배웠었고,

2주차에 JQuery, JSON 데이터, AJAX, OpenAPI를 통한 서버-클라이언트 통신방법에 대해 학습했다.

3주차에 파이썬, MongoDB, 웹 스크롤링 등 서버 프로그래밍에 대해 다루었다.

앞으로 4,5주차에 1,2,3주에 학습했던 내용을 종합한다고 한다.

4주차가 기대가 된다. Go!