

PJT명	Python을 활용한 데이터 수집 1	
단계	[Python PJT]	
진행일자	2025.07.25	
예상 구현 시간	필수기능	5H
	추가기능	2H
	심화기능	1H

1. 목표

- Python 기초 문법에 대하여 이해하고 활용할 수 있다.
- 파일 입출력에 대하여 이해한다.
- JSON 파일로 주어진 데이터를 프로그램에 활용할 수 있다.
- 데이터 구조를 분석하고 재구성하여 프로그램에 활용할 수 있다.
- Python을 활용하여 데이터를 가공하고 JSON 형태로 구성할 수 있다.

2. 준비사항

1) 프로젝트 구조

- 프로젝트는 총 두개의 폴더로 구성되어 있으며, 각 폴더별로 사용하는 데이터가 다름
 1. aladin
 2. spotify
- 각각 폴더에는 다음의 내용이 제공됨
 1. problem_*.py: 요구사항 구현을 위한 스켈레톤 코드
 2. examples 폴더: 아래 2) 사용 데이터에 설명된 examples 폴더
 3. data 폴더: 아래 2) 사용 데이터에 설명된 data 폴더

2) 사용 데이터

- examples 폴더: 이번 프로젝트 해결을 위해 알아야 하는, 혹은

직접적인 도움이 될 수 있는 코드

예시:

▼ EXAMPLES

```
01_example.py
02_example.py
03_example.py
sample.json
```

```
# 파일 열기
# open('파일경로', encoding='인코딩방식')
# - current_dir / 'sample.json': Path 객체의 / 연산자로 경로 결합
# - encoding="utf-8"은 한글 등 유니코드 문자를 올바르게 처리하기 위한 설정
book = open(current_dir / 'sample.json', encoding='utf-8')
```

- data 폴더: 실제 문제 풀이에 사용되는 데이터

예시:

▼ DATA

```
> books
book.json
books.json
categories.json
```

```
{
  "title": "노인과 바다",
  "link": "http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=12717326",
  "author": "어니스트 헤밍웨이 (지은이), 김육동 (옮긴이)",
  "description": "노벨 문학상, 풀리처상 수상 작가, 20세기 미국 문학의 거장",
  "isbn": "8937462788",
  "isbn13": "9788937462788",
  "id": 14522431,
  "priceSales": 7200,
  "priceStandard": 8000,
```

3) 개발언어 및 툴

- Python 3.11+ / Visual Studio Code

4) 필수 라이브러리 / 오픈소스

- Python 내장 json 모듈

3.작업 순서

- 1) 팀원과 같이 요구사항(기본/추가/심화)을 확인하고, GitLab에 프로젝트를 생성한다.
 - 프로젝트 이름은 01-pjt로 지정한다.
 - 각 반 담당 강사님을 Maintainer로 설정한다.
- 2) 제공된 examples의 코드를 확인하고, 요구사항에 필요한 코드를 파악한다.
- 3) 각 문제 코드를 확인하고, data에 주어진 파일을 활용하여 필수 요구사항을 구현한다.
- 4) 작성한 코드들을 정리하고, README를 작성한다.
- 5) README 작성이 완료되면 도전 과제를 진행한다.
- 6) 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 한다.

4. 요구사항

추천 알고리즘을 통한 영화 추천 커뮤니티 서비스를 구축하려고 한다. 다양한 스트리밍 플랫폼에서 제공되는 영화 정보를 수집 및 관리하고, 이를 기반으로 개인화된 영화 추천, 장르별 영화 탐색, 유사 영화 추천 등 다채로운 추천 기능을 설계 및 구현한다. 또한 영화에 대한 사용자 리뷰 및 감상평 공유 커뮤니티 기능을 제공하여, 사용자들이 활발하게 소통하고 정보를 교환할 수 있는 기능을 제공한다. 사용자는 자신이 본 영화를 평가하고, 다른 사용자의 리뷰를 참고하여 다음 영화를 선택하는데 도움을 받을 수 있다. 나아가, 관심 영화 목록을 맞춤형으로 구성하는 등 다양한 편의 기능을 제공한다. 팀원과 상의하여 아래 요구사항을 만족할 수 있도록 요구 사항 명세서를 작성 및 구현해보자.

본격적인 서비스 구현에 앞서서 제공되는 데이터 파일을 기반으로, 필요한 정보를 파싱한 후 프로그램 내부에서 정리할 수 있는 프로그램을 구현해보자.

- 요구사항 예시(참고용)
 - 아래의 내용을 참고하여 추가적인 아이디어에 대해 요구사항을 추가 또는 수정하여 기능을 구현한다. 단, **필수 기능은 구현해야 하며, 수정할 수 없다.**

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F01	도서 데이터	데이터 수집	주어진 파일에서 필요한 정보를 추출하는 기능	필수
F02	도서 데이터	데이터 변환	F01에서 추출한 정보를 재사용 가능하게 변환하는 기능	필수
F03	도서 데이터	데이터 수정	추출한 데이터를, 다른 데이터를 바탕으로 수정하는 기능	필수

F04	도서 데이터	다중 데이터 분석	여러 개체에 대한 정보가 주어진 파일에서 각각의 필요한 정보를 추출하는 기능	필수
F05	도서 데이터	다중 데이터 변환	F04에서 추출한 다중 데이터를 재사용 가능하게 변환하는 기능	필수
F06	도서 데이터	다중 파일 데이터 수집	여러 파일에 걸쳐서 데이터를 추출하는 기능	필수
F07	도서 데이터	특정 데이터 출력	F06에서 추출한 데이터 중 특정 조건을 만족하는 데이터를 출력하는 기능	필수
F08	도서 데이터	특정 데이터 수집	F06에서 추출한 데이터 중 특정 조건을 만족하는 데이터를 재사용 가능하게 변환하는 기능	필수
F09	도서 데이터	특정 데이터 수집	생성형 AI를 활용하여 특정 조건을 만족하는 데이터를 추출하는 기능	도전
F10	도서 데이터	특정 데이터 수집 및 정리	생성형 AI를 활용하여 특정 조건을 만족하는 데이터를 추출하고 정리하는 기능	도전
F11	음악 데이터	데이터 수집	주어진 파일에서 필요한 정보를 추출하는 기능 id, name, genres_ids, images, type	필수
F12	음악 데이터	데이터 변환	F11에서 추출한 정보를 재사용 가능하게 변환하는 기능	필수
F13	음악 데이터	데이터 수정	추출한 데이터를, 다른 데이터를 바탕으로 수정하는 기능	필수
F14	음악 데이터	다중 데이터 분석	여러 개체에 대한 정보가 주어진 파일에서 각각의 필요한 정보를 추출하는 기능	필수
F15	음악 데이터	다중 데이터 변환	F14에서 추출한 다중 데이터를 재사용 가능하게 변환하는 기능	필수
F16	음악 데이터	다중 파일 데이터 수집	여러 파일에 걸쳐서 데이터를 추출하는 기능	필수
F17	음악 데이터	특정 데이터 출력	F16에서 추출한 데이터 중 특정 조건을 만족하는 데이터를 출력하는 기능	필수
F18	음악 데이터	특정 데이터 수집	F16에서 추출한 데이터 중 특정 조건을 만족하는 데이터를 재사용 가능하게 변환하는 기능	필수

F19	음악 데이터	특정 데이터 수집	생성형 AI를 활용하여 특정 조건을 만족하는 데이터를 추출하는 기능	도전
F20	음악 데이터	특정 데이터 수집	생성형 AI를 활용하여 특정 조건을 만족하는 데이터를 추출하는 기능	도전
...

1) 기본(필수) 기능 (aladin)

A. 제공되는 도서 데이터의 주요내용 수집

book.json 파일에서 필요한 정보에 해당하는 값을 추출하여, 새로운 Dictionary로 반환하는 함수 book_info를 작성한다.

- 요구사항 번호: F01, F02
- 주어진 aladin/problem_a.py를 수정하여 구현
- 필요한 정보: id, name, author, priceSales, description, cover, categoryId
- 구현해야 할 화면 (콘솔 출력)

```
{'author': '어니스트 헤밍웨이 (지은이), 김옥동 (옮긴이)',  
'categoryId': [151128, 50919],  
'cover': 'https://image.aladin.co.kr/product/1452/24/coversum/8937462788_2.jpg',  
'description': '노벨 문학상, 풀리처상 수상 작가, 20세기 미국 문학을 개척한 작가 어니스트 헤밍웨이의 대표작.  
"문학의 개척자라 불리는 헤밍웨이는 제1차 세계대전 후 삶의 좌표를 잃어버린 '길 잃은 세대'를 대  
"작가이다. '민음사 세계문학전집' 278권으로 출간된 <노인과 바다>는 헤밍웨이의 마지막 소설  
'작가 고유의 소설 수법과 실존 철학이 짧은 분량 안에 집약되어 있다.',  
'id': 14522431,  
'priceSales': 7200,  
'title': '노인과 바다'}
```

- 완성된 함수는 다음 문제의 기본 기능으로 사용됨

B. 제공되는 도서 데이터의 주요내용 수정

book.json 파일에서 필요한 정보에 해당하는 값을 추출한 다음, categoryId는 categories.json 파일에서 각각 ID에 매칭되는 categoryName으로 변환하여 새로운 Dictionary로 반환하는 함수 book_info를 작성한다.

- 요구사항 번호: F03
- 요구사항 A의 코드를 활용하여 주어진 aladin/problem_b.py를 수정하여 구현
- 필요한 정보: id, name, author, priceSales, description, cover, categoryName
- 구현해야 할 화면 (콘솔 출력)

```
{'author': '어니스트 헤밍웨이 (지은이), 김옥동 (옮긴이)',  
'categoryName': ['문학', '영미소설'],  
'cover': 'https://image.aladin.co.kr/product/1452/24/coversum/8937462788_2.jpg',  
'description': '노벨 문학상, 풀리처상 수상 작가, 20세기 미국 문학을 개척한 작가 어니스트 헤밍웨이. "문학의 개척자라 불리는 헤밍웨이는 제1차 세계대전 후 삶의 좌표를 잃어버린 '길'을 찾아 떠나는 작가이다. '민음사 세계문학전집' 278권으로 출간된 <노인과 바다>는 헤밍웨이의 대표작이자, 작가 고유의 소설 수법과 실존 철학이 짧은 분량 안에 집약되어 있다.',  
'id': 14522431,  
'priceSales': 7200,  
'title': '노인과 바다'}
```

- 예시 화면의 경우 pprint 모듈을 활용하여 출력되어 Dictionary의 key 순서가 정렬되어서 출력
- 완성된 함수는 다음 문제의 기본 기능으로 사용됨

C. 다중 데이터 분석 및 수정

books.json 파일에는 여러 도서에 대한 데이터가 제공된다. 각각 도서에 대한 데이터 중 필요한 정보를 추출하여 Dictionary로 변환한 다음, 해당 Dictionary 들을 담고 있는 새로운 List를 반환하는 함수 books_info를 작성한다.

- 요구사항 번호: F04, F05
- 요구사항 B의 코드를 활용하여 주어진 aladin/problem_c.py를 수정하여 구현
- 필요한 정보: id, name, author, priceSales, description, cover, categoryName
- 구현해야 할 화면 (콘솔 출력, 결과 일부 생략)

```
[{'author': '제인 오스틴 (지은이), 윤지관, 전승희 (옮긴이)',  
  'categoryName': ['문학', '영미소설'],  
  'cover': 'https://image.aladin.co.kr/product/43/68/coversum/8937460882_3.jpg',  
  'description': '제인 오스틴의 대표작 <오만과 편견>이 보다 정확하고 말끔한 번역으로 재출간됐다. 역자인 윤지관  
  '10여 년에 걸친 기간 동안 철저한 원문대조를 통해, 본래의 의미와 문체를 생생하게 되살려내기 위  
  '노력했다고.',  
  'id': 436838,  
  'priceSales': 11700,  
  'title': '오만과 편견'},  
 {'author': '어니스트 헤밍웨이 (지은이), 김옥동 (옮긴이)',  
  'categoryName': ['문학', '영미소설'],  
  'cover': 'https://image.aladin.co.kr/product/1452/24/coversum/8937462788_2.jpg',  
  'description': '노벨 문학상, 풀리처상 수상 작가, 20세기 미국 문학을 개척한 작가 어니스트 헤밍웨이의 대표작.  
  "문학의 개척자라 불리는 헤밍웨이는 제1차 세계대전 후 삶의 좌표를 잃어버린 '길 잃은 세대'를 대  
  "작가이다. '민음사 세계문학전집' 278권으로 출간된 <노인과 바다>는 헤밍웨이의 마지막 소설  
  "작가 고유의 소설 수법과 실존 철학이 짧은 분량 안에 집약되어 있다.',  
  'id': 14522431,  
  'priceSales': 7200,  
  'title': '노인과 바다'},  
 {'author': '서머셋 모음 (지은이), 송무 (옮긴이)',  
  'categoryName': ['문학', '영미소설'],  
  'cover': 'https://image.aladin.co.kr/product/23/77/coversum/s692639624_1.jpg',  
  'description': '<달과 6펜스>는 15종에 이르는 번역본이 이미 소개되어 있을 만큼 국내에서 크게 환영받는 작품이다  
  '작품은 서머셋 모음 전세계에 널리 알려진 결정적인 작품으로 제1차 세계대전이 끝난 이듬해인 1919년  
  '출판되어 대단한 인기를 끌었다.',  
  'id': 237763,  
  'priceSales': 9000,  
  'title': '달과 6펜스'},
```

D. 가장 리뷰가 높은 도서 데이터 수집

books 폴더에는 <도서id>.json 형식으로 각 도서에 대한 정보가 제공되며, 도서의 회원 리뷰 평점은 customerReviewRank에 기록되어 있다. books.json 파일에 제공된 각 도서 정보의 id를 활용하여 해당 도서의 리뷰 평점 정보를 확인하고, 이 중 리뷰 평점이 가장 높은 도서의 제목을 출력하는 함수 best_book을 작성한다.

- 요구사항 번호: F06, F07
- 주어진 aladin/problem_d.py를 수정하여 구현
- 반복문을 통해 aladin/data/books 폴더 내부의 파일들을 열기
 - 주어진 aladin/examples/03_example.py의 코드를 참고하여 작성
- 구현해야 할 화면 (콘솔 출력)

멋진 신세계

E. 특정 년도에 출판한 도서 데이터 수집

books 폴더에는 <도서id>.json의 형식으로 각 도서에 대한 정보가 제공되며, 도서의 출판 연도는 pubDate에 포함되어 있다. books.json 파일에 제공된 각 도서 정보의 id를 활용하여, 해당 도서의 출판 연도를 확인하고, 출판 연도가 2023년인 도서들의 제목을 리스트로 출력하는 함수 new_books를 작성한다.

- 요구사항 번호: F08
- 주어진 aladin/problem_e.py를 수정하여 작업
- 반복문을 통해 aladin/data/books 폴더 내부의 파일들을 열기
 - 주어진 03_example.py의 코드를 참고하여 작성
- 구현해야 할 화면 (콘솔 출력)

```
['회복탄력성의 힘 - 쉽게 포기하지 않고 결국 해내는 아이의 비밀',  
'사피엔스 - 유인원에서 사이보그까지, 인간 역사의 대담하고 위대한  
질문']
```

2) 도전 과제 (aladin)

기본 기능 구현 후 생성형 AI를 사용해 다음 두 도전과제 요구사항을 해결한다.

A. 2023년 출판 도서 중 회원 리뷰 평점이 가장 높은 도서의 제목을 수집하는 함수 `best_new_books`를 작성한다.

- 요구사항 번호: F09
- 주어진 `aladin/problem_f_1.py`를 수정하여 작업
- 실행 예시 (콘솔 출력)

사피엔스 - 유인원에서 사이보그까지, 인간 역사의 대담하고 위대한 질문

B. 카테고리가 컴퓨터 공학인 도서의 제목을 판매가격(`priceSales`)이 높은 순서대로 수집하는 함수 `sorted_cs_books_by_price`를 작성한다.

- 요구사항 번호: F10
- 주어진 `aladin/problem_f_2.py`를 수정하여 작업
- 실행 예시 (콘솔 출력)

['한 권으로 읽는 컴퓨터 구조와 프로그래밍 - 더 나은 소프트웨어 개발을 위한 하드웨어, 자료구조, 필수 알고리즘 등 프로그래머의 비밀 노트', '클린 코드 Clean Code - 애자일 소프트웨어 장인 정신']

3) 기본(필수) 기능 (spotify)

A. 제공되는 아티스트 데이터의 주요내용 수집

artist.json 파일에서 필요한 정보에 해당하는 값을 추출하여, 새로운 Dictionary로 반환하는 함수 artist_info를 작성한다.

- 요구사항 번호: F11, F12
- 주어진 spotify/problem_a.py를 수정하여 작업
- 필요한 정보: id, name, genres_ids, images, type
- 구현해야 할 화면 (콘솔 출력)

```
{'genres_ids': [651, 816],  
  'id': 178,  
  'images': [{'height': 640,  
              'url': 'https://i.scdn.co/image/ab6761610000e5eb59f8cfc8e71dcaf8c6ec4bde',  
              'width': 640},  
             {'height': 320,  
              'url': 'https://i.scdn.co/image/ab6761610000517459f8cfc8e71dcaf8c6ec4bde',  
              'width': 320},  
             {'height': 160,  
              'url': 'https://i.scdn.co/image/ab6761610000f17859f8cfc8e71dcaf8c6ec4bde',  
              'width': 160}],  
  'name': 'Jimin',  
  'type': 'artist'}
```

- 완성한 함수는 다음 문제의 기본 기능으로 사용됨

B. 제공되는 아티스트 데이터의 주요내용 수정

artist.json 파일에서 필요한 정보에 해당하는 값을 추출한 다음,
genres_ids는 genres.json 파일에서 각각 ID에 매칭되는
genre_name으로 변환하여 genres_names로 수집한 뒤 새로운
Dictionary로 반환하는 함수 artist_info를 작성한다.

- 요구사항 번호: F13
- 요구사항 A의 코드를 활용하여 주어진 spotify/problem_b.py를 수정
- 필요한 정보: id, name, images, type, genre_names
- 구현해야 할 화면 (콘솔 출력)

```
{'genres_names': ['punk-rock', 'anime'],  
 'id': 178,  
 'images': [{ 'height': 640,  
               'url': 'https://i.scdn.co/image/ab6761610000e5eb59f8cfc8e71dcaf8c6ec4bde',  
               'width': 640},  
             { 'height': 320,  
               'url': 'https://i.scdn.co/image/ab6761610000517459f8cfc8e71dcaf8c6ec4bde',  
               'width': 320},  
             { 'height': 160,  
               'url': 'https://i.scdn.co/image/ab6761610000f17859f8cfc8e71dcaf8c6ec4bde',  
               'width': 160}],  
 'name': 'Jimin',  
 'type': 'artist'}
```

- 예시 화면의 경우 pprint 모듈을 활용하여 출력되어 Dictionary의
key 순서가 정렬되어서 출력
- 완성된 함수는 다음 문제의 기본 기능으로 사용됨

C. 다중 데이터 분석 및 수정

artists.json 파일에는 여러 아티스트에 대한 데이터가 제공된다. 각각 아티스트에 대한 데이터 중 필요한 정보를 추출하여 Dictionary로 변환한 다음, 해당 Dictionary 들을 담고 있는 새로운 List를 반환하는 함수 artist_info를 작성한다.

- 요구사항 번호: F14, F15
- 요구사항 B의 코드를 활용하여 주어진 spotify/problem_c.py를 수정
- 필요한 정보: id, name, images, type, genres_names
- 구현해야 할 화면 (콘솔 출력)

```
[{'genres_names': ['acoustic', 'electro', 'j-rock'],
  'id': 451,
  'images': [{'height': 640,
    'url': 'https://i.scdn.co/image/ab6761610000e5ebd642648235ebf3460d2d1f6a',
    'width': 640},
    {'height': 320,
    'url': 'https://i.scdn.co/image/ab67616100005174d642648235ebf3460d2d1f6a',
    'width': 320},
    {'height': 160,
    'url': 'https://i.scdn.co/image/ab6761610000f178d642648235ebf3460d2d1f6a',
    'width': 160}],
  'name': 'BTS',
  'type': 'artist'},
{'genres_names': ['edm'],
  'id': 116,
  'images': [{'height': 640,
    'url': 'https://i.scdn.co/image/ab6761610000e5eb6199c3c2f414880e2b9077a9',
    'width': 640},
    {'height': 320,
    'url': 'https://i.scdn.co/image/ab676161000051746199c3c2f414880e2b9077a9',
    'width': 320},
    {'height': 160,
    'url': 'https://i.scdn.co/image/ab6761610000f1786199c3c2f414880e2b9077a9',
    'width': 160}],
  'name': 'NewJeans',
  'type': 'artist'}]
```

D. 가장 인기가 높은 아티스트 데이터 수집

artists 폴더에는 <아티스트id>.json 형식으로 각 아티스트에 대한 정보가 제공되어 있으며, 아티스트의 인기도는 popularity에 기록되어 있다. artists.json 파일에 제공된 각 아티스트 정보의 id를 활용하여, 해당 아티스트의 인기를 확인하고, 이 중 인기가 가장 높은 아티스트의 이름을 출력하는 함수 max_popularity를 작성한다.

- 요구사항 번호: F16, F17
- 주어진 spotify/problem_d.py를 수정하여 구현
- 반복문을 통해 spotify/data/artists 폴더 내부의 파일들을 열기
 - 주어진 spotify/examples/03_example.py의 코드를 참고하여 작성
- 구현해야 할 화면 (콘솔 출력)

NewJeans

E. 특정 팔로워 수 이상의 아티스트 데이터 수집

artists 폴더에는 <아티스트id>.json 형식으로 각 아티스트에 대한 정보가 제공되어 있으며, 아티스트의 팔로워 수는 followers에 기록되어 있다. artists.json 파일에 제공된 각 아티스트 정보의 id를 활용하여, 해당 아티스트의 팔로워 수를 확인하고, 팔로워 수의 총합이 10,000,000 이상인 아티스트 이름과 URI를 리스트로 출력하는 함수 dec_artists를 작성한다.

- 요구사항 번호: F18
- 주어진 spotify/problem_e.py를 수정하여 작업
- 반복문을 통해 spotify/data/artists 폴더 내부의 파일들을 열기
 - 주어진 spotify/examples/03_example.py의 코드를 참고하여 작성
- 구현해야 할 화면 (콘솔 출력)

```
[{'name': 'BTS', 'uri-id': '3Nrfpe0tUJi4K4DXYWgMUX'},  
{ 'name': 'BLACKPINK', 'uri-id': '41MozSoPIsD1dJM0CLPjZF'},  
{ 'name': 'Stray Kids', 'uri-id': '2dIgFjalVxs4ThymZ67YCE'},  
{ 'name': 'j-hope', 'uri-id': '0b1sIQumIAsNbqAoIClSpy'}]
```

4) 도전 과제 (spotify)

기본 기능 구현 후 생성형 AI를 사용해 다음 두 도전과제 요구사항을 해결한다.

A. 팔로워가 5,000,000 이상, 10,000,000 미만인 아티스트들의 이름과 팔로워 수를 수집하는 함수 `get_popular`를 작성한다

- 요구사항 번호: F19
- 주어진 `spotify/problem_f_1.py`를 수정하여 작업
- 실행 예시 (콘솔 출력)

```
[{'followers': 5901644, 'name': 'Jimin'},  
{ 'followers': 9571638, 'name': 'SEVENTEEN'},  
{ 'followers': 8041766, 'name': 'IU'},  
{ 'followers': 8000368, 'name': 'Jung Kook'},  
{ 'followers': 6278033, 'name': '(G)I-DLE'},  
{ 'followers': 6785638, 'name': 'NCT DREAM'}]
```

B. 장르에 `acoustic`이 포함된 아티스트의 이름을 수집하는 함수 `acoustic_artists` 함수를 작성한다.

- 요구사항 번호 F20
- 주어진 `spotify/problem_f_2.py`를 수정하여 작업
- 실행 예시 (콘솔 출력)

```
['BTS']
```

상기된 과제를 전부 완료한 경우 명시된 요구사항 외 추가 개발을 자유롭게 진행한다.

5. 참고자료

- json – JSON encoder and decoder
<https://docs.python.org/3.11/library/json.html>
- pprint - Data pretty printer
<https://docs.python.org/3.11/library/pprint.html>

6. 결과

제출 기한은 진행일 18시까지이므로 제출 기한을 지킬 수 있도록 한다. 제출은 GitLab을 통해서 이뤄진다.

- 산출물과 제출
 - 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
 - 완성된 각 문제 별 소스코드 및 실행 화면 캡처본
 - 프로젝트 이름은 01-pjt로 지정, 각자의 계정에 생성할 것
 - 각 반 담당 강사님을 Maintainer로 설정

- 끝 -