파이썬을 활용한 데이터 수집

1. 목표

- 기초 Python에 대한 이해
- Python을 통한 데이터 수집 및 파일 저장
- Python 조건/반복문 및 다양한 자료구조 조작
- API 활용을 통한 데이터를 수집 및 가공

2. 준비 사항

1. Python 환경 설정
    - python 3.7 이상
    - `Visual Studio Code`
2. 필수 라이브러리 활용
    - requests
3. 필수 API
    - 영화진흥위원회 오픈 API
        - API 활용 시, 요청 URL 중 `json` 버전을 사용
        - 주간/주말 박스오피스 API 서비스
        - 영화 상세정보 API 서비스

[주의] API키는 반드시 환경 변수에 저장하여 사용하세요. 절대 소스 코드에 직접 입력하지 마세요.

3. 요구 사항

- 영화 관련 서비스를 만들기 위한 데이터 수집 단계로, 영화 데이터베이스 구축을 위한 csv 파일을 작성합니다.

1. **영화진흥위원회 오픈 API (주간/주말 박스오피스 데이터) - `boxoffice.py`**
    - 최근 50주간 데이터 중에 주간 박스오피스 TOP10데이터를 수집합니다. 해당 데이터는 향후 영화 관련 서비스에서 기본으로 제공되는 영화 목록으로 사용될 예정입니다.
        - 요청 조건
            1. **주간(월~일)** 기간의 데이터를 조회합니다.
            2. 조회 기간은 총 **50주**이며, 기준일(마지막 일자)은 **2020년 5월 31일**입니다.
            3. 다양성 영화/상업 영화를 모두 포함하여야 합니다.
            4. 한국/외국 영화를 모두 포함하여야 합니다.
            5. 모든 상영지역을 포함하여야 합니다.

- 결과
  - 수집된 데이터에서 `영화 대표코드`, `영화명`, `누적관객수`를 기록합니다.
  - `누적관객수`는 중복시 최신 정보를 반영하여야 합니다.
    예) 영화 아쿠아맨이 20190113 기준 50,000명이고, 20190106 기준 5,000명이면 50,000명이 저장되어야 합니다.
  - 해당 결과를 **boxoffice.csv**에 저장합니다.

2. **영화진흥위원회 오픈 API (영화 상세정보) - `movies.py`**
   - 위에서 수집한 `영화 대표코드`를 활용하여 상세 정보를 수집합니다. 해당 데이터는 향후 영화 관련 서비스에서 영화 정보로 활용될 것입니다.
   - 결과
     - 영화별로 다음과 같은 내용을 저장합니다.

       `영화 대표코드`, `영화명(국문)`, `영화명(영문)`, `영화명(원문)`, `관람등급`, `개봉연도`, `상영시간`, `장르`, `감독명`
     - 해당 결과를 **movies.csv**에 저장합니다.

4. 결과

- `boxoffice.py`, `boxoffice.csv`, `movies.py`, `movies.csv`까지 총 4개의 파일을 저장합니다.
- 결과물은 개인 GitHub의 **TIL**에 Python > data-collection 폴더를 생성하고 해당 폴더에 파일들을 저장하여 업로드 합니다.