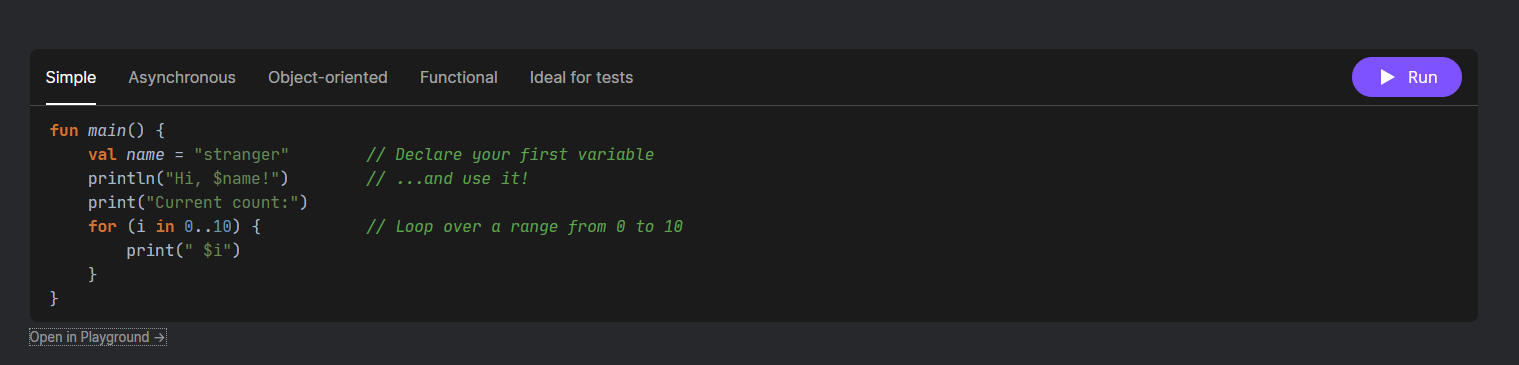
1. **Try**

* 머신러닝에 대한 관심
  + <https://wikidocs.net/book/2155>
  + <https://wikidocs.net/book/2891>
* 파이썬에 대한 관심
  + <https://wikidocs.net/book/2>
* 도커에 대한 관심
  + <https://opentutorials.org/course/128/8657>
* K8S에 대한 관심
  + <https://velog.io/@ks0624/%EB%94%B0%EB%9D%BC%ED%95%98%EB%A9%B0-%EB%B0%B0%EC%9A%B0%EB%8A%94-%EC%BF%A0%EB%B2%84%EB%84%A4%ED%8B%B0%EC%8A%A4-%EC%9E%85%EB%AC%B81>
* 코틀린 언어에 대한 관심
  + <https://m.boostcourse.org/mo132/lectures/28611>
  + <https://kotlinlang.org/> 사이트의 예제와 Playground 사이트에서 연습



* 코틀린의 코루틴에 대한 강의

<https://www.youtube.com/watch?v=Vs34wiuJMYk&list=PLbJr8hAHHCP5N6Lsot8SAnC28SoxwAU5A>

1. **Done**

* 영화의 취향 → 국가를 넘어서 음악 추천 / 저작권에 대한 이해와 활용
* 영화의 취향 → 장르를 넘어서 음악 추천
* 영화의 선택(하나의 영화 선택) → 영화의 취향

1. **To Do**

* Object Diagram ( 시스템 참여자 식별 )
  + 영화 정보
    - 영화 리스트
      * <https://www.kobis.or.kr/kobisopenapi/homepg/apiservice/searchServiceInfo.do>
      * 제작년도 검색
        + <https://www.kobis.or.kr/kobisopenapi/webservice/rest/movie/searchMovieList.json?key=f754bc71a8129e908e1455927354f2bf&prdtStartYear=2000&prdtEndYear=2001>
      * 국적 코드 검색
        + <http://www.kobis.or.kr/kobisopenapi/webservice/rest/code/searchCodeList.json?key=f754bc71a8129e908e1455927354f2bf&comCode=220310>
      * 제작년도+국적 검색(대만영화)
        + <https://www.kobis.or.kr/kobisopenapi/webservice/rest/movie/searchMovieList.json?key=f754bc71a8129e908e1455927354f2bf&prdtStartYear=2000&prdtEndYear=2001&repNationCd=22041001>
      * 영화 분류 코드 검색
        + <http://www.kobis.or.kr/kobisopenapi/webservice/rest/code/searchCodeList.json?key=f754bc71a8129e908e1455927354f2bf&comCode=2201>
      * 제작년도 + 영화 분류 코드 검색(단편영화)
        + <https://kobis.or.kr/kobisopenapi/webservice/rest/movie/searchMovieList.json?key=f754bc71a8129e908e1455927354f2bf&prdtStartYear=2000&prdtEndYear=2001&movieTypeCd=220102>
    - 영화 상세 정보
      * <https://www.kobis.or.kr/kobisopenapi/webservice/rest/movie/searchMovieInfo.json?key=f754bc71a8129e908e1455927354f2bf&movieCd=20230174>
* Sprint 1 에 대한 계획
  + Spring Planning : 무엇을 하고, 기간?
    - 기간
      * API-1 : 될 때까지….
      * API-2 : 될 때까지….
      * API-3 : 될 때까지….
    - Python Flask - API 프레임 워크
    - 추천 영화 정보를 제공해 주는 API 개발
      * 영화 선택 → 영화 취향을 결과 전송
        + **마음속에 ‘????’ 정했어 → 조건 : 영화 제목 검색 (API-1)**

1. 검색어 입력
2. 영화 검색 결과 조회
3. 영화 선택
4. 영화 취향 결과 조회
   1. ex: **타이타닉 선택 → 상세 정보(장르)** → \* 취향 → 음악과 연관된….
      * + - 마음속에 ‘????’ 못 정했어 → 조건 :

영화 제목 검색

**제작년도 (API-2)**

시대별

미리 시대별 영화 분류된 데이터를 만들던지, 수집

나라

영화 유형

* + - * + 제공되는 영화 후보 (????? 추출 / 최초 추천)

후보 : 많이 검색 / 선택 한 영화

**랜덤 (API-3)**

상황인지(Context Recognition)

* + - * 오픈 API 구조
        + 영화 리스트 검색 → movieCd
        + 영화 상세 정보 검색 → movieCd 사용
      * 한국영화 데이터 베이스
        + 가입 : <https://www.kmdb.or.kr/>
        + 인증키 발급 : 2~3일 소요
      * 한국 영화 진흥 위원회
        + 가입 : <https://www.kobis.or.kr/>
        + 인증키 발급 : 즉시 발급
* WBS 일부 작성
* 개발 환경
  + 파이썬 설치
  + Pycharm 설치
  + Docker 설치
  + pip install flask (API)
  + 코드 작성
  + Console 실행
  + Docker 실행
* 분석 환경 (개발 환경 완료 후)
  + 머신러닝 환경 (pip install …..)
    - numpy
    - pandas
    - sklearn
  + 데이터 수집
    - API 기반의 데이터 수집 개발
  + 모형 설계 / 개발
    - 알고리즘 선택 및 모형 개발
    - X 변수들… ( 선택한 영화의 요소들 )
    - Y 변수 ( 음악 취향 )

1. **Concept**

* Architecture(\*)