SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

01 PJT

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

파이썬을 활용한 API 데이터 수집

챕터의 포인트

- [도전] 날씨 데이터 수집
- [심화] 금융 데이터 수집
- 제출

SAMSUNG SW ACADEMY FOR YOUTH

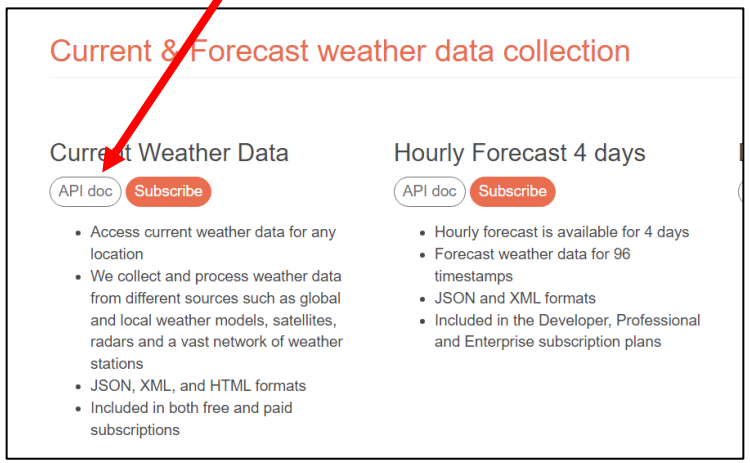
함께가요 미래로!
Enabling People

[도전]
날씨 데이터 수집

공통 요구사항

OpenWeather

- 외부 API 를 사용하여 데이터를 받아오고, 데이터를 원하는 형태로 가공하는 과정입니다.
- 요구사항에서 사용할 API 는 OpenWeatherMap API 입니다.
  - 공식 문서를 보고 데이터의 구조를 먼저 파악한 후 요구사항을 구현합니다.
  - 공식문서에서 제공하는 API 중 Current Weather Data 를 이용합니다.



세부 요구사항

A. 데이터 추출 - KEY값 출력하기

B. 데이터 추출 - 원하는 값만 추출하기

C. 데이터 가공 - KEY값 변경하기

D. 데이터 가공 - 데이터 추가하기

E. 생성형 AI 활용하기

- [참고] 모든 출력 결과는 예시입니다. 날짜에 따라 다르게 나올 수 있습니다.

A. 데이터 추출 - Key 값 출력하기

- 날씨 데이터의 응답을 json 형태로 변환 후 아래와 같이 Key 값만 출력하도록 구성합니다.
- 공식 문서의 요청 변수 및 예제 요청결과(JSON) 부분을 참고합니다.

- 정답 예시

```
dict_keys(['coord', 'weather', 'base', 'main', 'visibility', 'wind',
'clouds', 'dt', 'sys', 'timezone', 'id', 'name', 'cod'])
```

B. 데이터 추출 - 원하는 값만 추출하기

- 날씨 데이터 중 다음 조건에 해당하는 값만 딕셔너리 형태로 반환하는 함수를 구성합니다.
  - KEY 값이“main” 인 데이터
  - KEY 값이 “weather” 인 데이터
  - 함수에서 두 데이터를 새로운 dictionary 에 담아서 return 합니다.
- 정답 예시

```
{'main': {'feels_like': 276,
          'humidity': 81,
          'pressure': 1026,
          'temp': 277.76,
          'temp_max': 278.91,
          'temp_min': 273.81},
 'weather_data': [{'description': 'clear sky',
                   'icon': '01d',
                   'id': 800,
                   'main': 'Clear'}]}
```

C. 데이터 가공 - KEY 값 변경하기

- B번에서 얻는 결과를 활용하여, KEY 값들을 한글로 변경한 딕셔너리를 반환하도록 구성합니다.

- 한글 키 값

```
feels_like : '체감온도',
humidity : '습도',
pressure : '기압',
temp : '온도',
temp_max : '최고온도',
temp_min : '최저온도',
description : '요약',
icon : '아이콘',
main : '핵심'
id : '식별자'
```

- 결과 예시

```
{'기본': {'기압': 1026,
        '습도': 81,
        '온도': 277.99,
        '체감온도': 276.27,
        '최고온도': 279.84,
        '최저온도': 274.81},
 '날씨': [{'식별자': 800, '아이콘': '01d',
 '요약': 'clear sky', '핵심': 'Clear'}]}
```

D. 데이터 가공 - 데이터 추가하기

- C번에서 얻는 결과를 활용하여, 섭씨 온도 데이터를 추가합니다.
  - 온도, 체감온도, 최저온도, 최고온도 4가지 데이터를 활용합니다.
- 결과 예시


```
{'기본': {'기압': 1026,
        '습도': 66,
        '온도': 281.54,
        '온도(섭씨)': 8.39,
        '체감온도': 279.12,
        '체감온도(섭씨)': 5.97,
        '최고온도': 282.91,
        '최고온도(섭씨)': 9.76,
        '최저온도': 276.81,
        '최저온도(섭씨)': 3.66},
 '날씨': [{'식별자': 800, '아이콘': '01d',
 '요약': 'clear sky', '핵심': 'Clear'}]}
```

E. 생성형 AI 활용하기

- 생성형 AI 를 활용하여 자유롭게 데이터를 수집하고, 추출하고 가공해봅니다
- 힌트
  - 내가 사용하는 API (OpenWeatherMap API)를 잘 알고 있는 지 확인합니다.
    - 질문 추천: OpenWeatherMap api 의 엔드포인트를 알려줘
      - 엔드포인트(Endpoint): 웹 API 에서 특정 기능이나 리소스에 접근할 수 있는 URL
  - 원하는 동작을 자유롭게 정하고, 해당 내용의 정답을 얻을 수 있도록 프롬프트를 구성합니다.
    - 질문 예시: 서울의 3일 후 날씨를 알기 위해서는 어떻게 요청을 보내야 해 ?
- 프롬프트 화면 캡처, 코드, 코드 실행 결과를 모두 제출합니다.

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

[심화]
금융 상품 데이터 수집

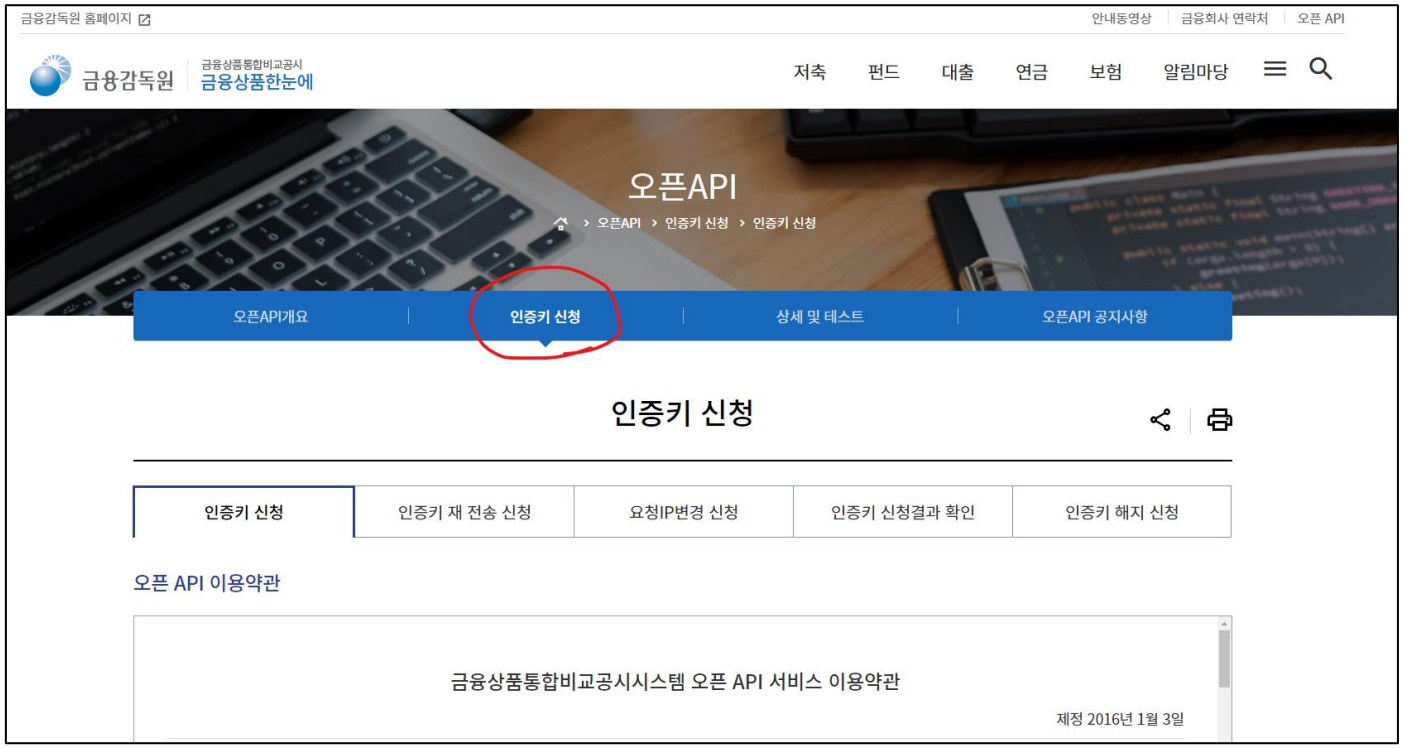
공통 요구사항

- 심화 도전 과제는 선택 사항입니다.
- 외부 API 를 사용하여 데이터를 받아오고, 데이터를 원하는 형태로 가공하는 과정입니다.

- 요구사항에서 사용할 API 는 금융상품통합비교공시 API 입니다.
  - 제공하는 API 중 정기예금 API 를 활용합니다.
  - 공식 문서를 보고 데이터의 구조를 먼저 파악한 후 요구사항을 구현합니다.

공통 요구사항 - API KEY 발급(1/4)

- 사이트 접속 및 인증키 신청 탭 클릭



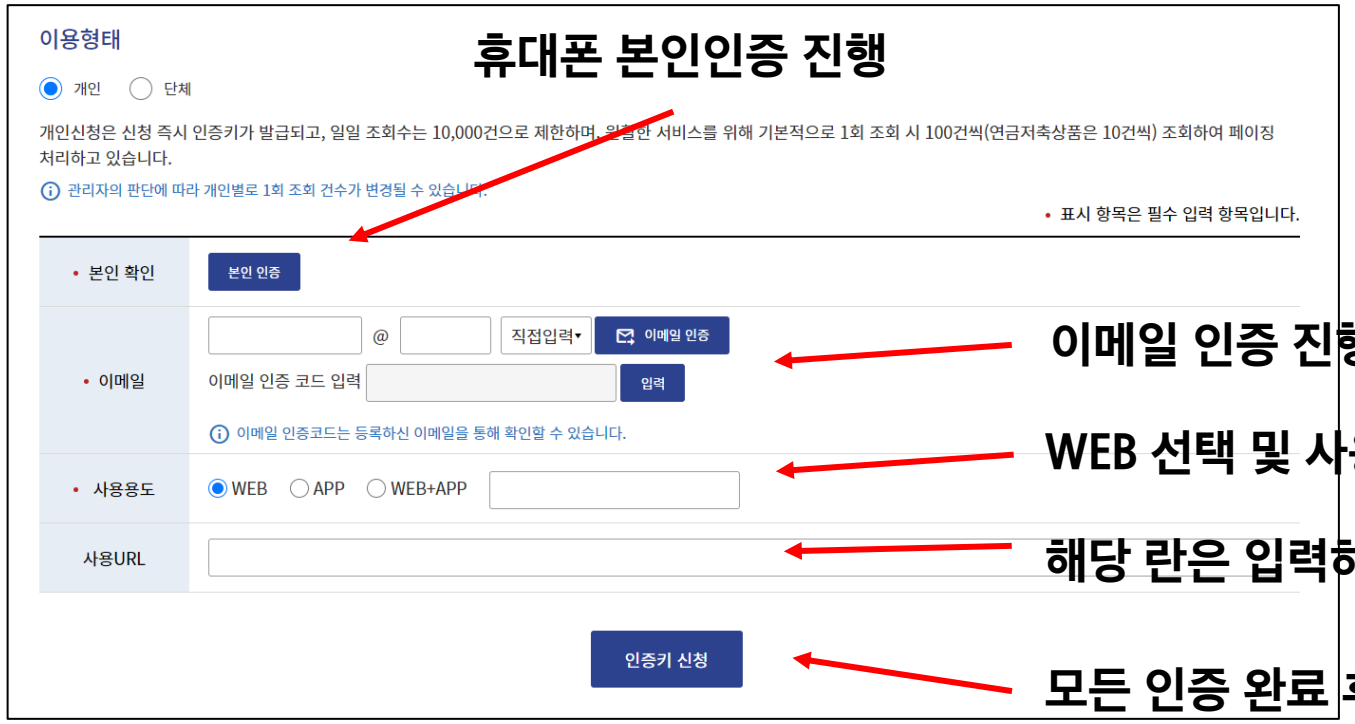
공통 요구사항 - API KEY 발급(2/4)

- 스크롤을 내려 하단의 이용약관에 동의합니다.

공통 요구사항 - API KEY 발급(3/4)

- 스크롤을 내려 하단의 신청 정보를 작성합니다.

개인 선택



휴대폰 본인인증 진행

이메일 인증 진행

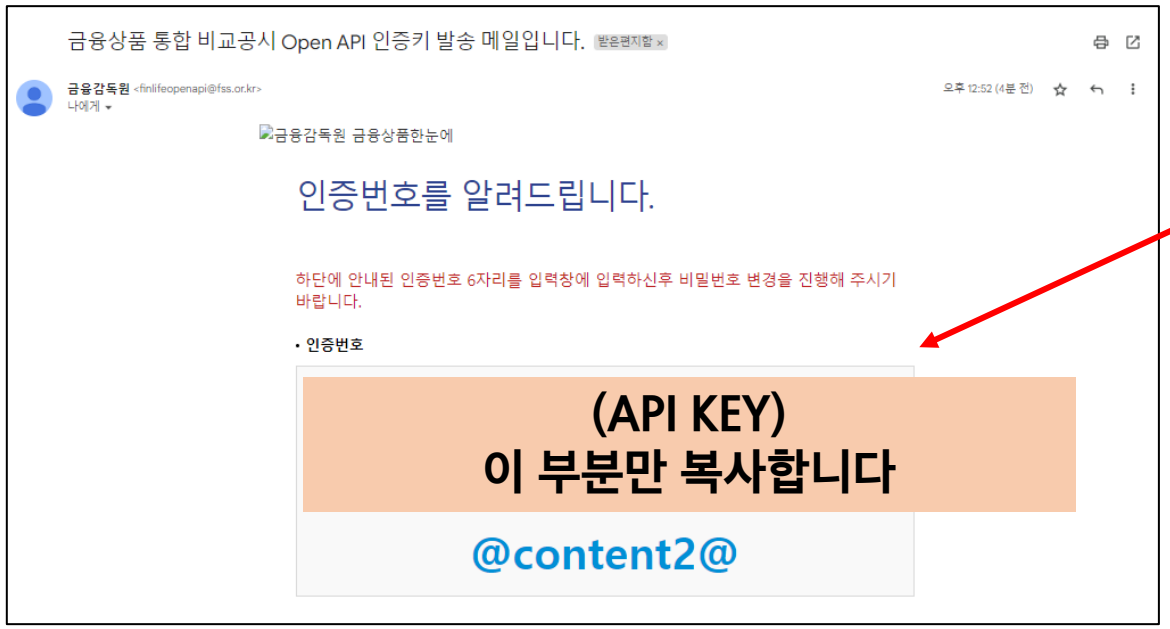
WEB 선택 및 사용용도 작성

해당 란은 입력하지 않습니다

모든 인증 완료 후 인증키 신청 버튼 클릭

공통 요구사항 - API KEY 발급(4/4)

- 인증한 메일로 전송된 인증키를 확인한 후 복사하여 활용합니다.



@content2@ 앞의 API KEY 에 해당하는
문자열만 복사하여 KEY 로 활용합니다

공통 요구사항 - API 사용 예제

```python
import pprint
import requests

def get_deposit_products():
    api_key = "MY_API_KEY"

    url = 'http://finlife.fss.or.kr/finlifeapi/depositProductsSearch.json'
    params = {
        'auth': api_key,
        # 금융회사 코드 020000(은행), 030200(여신전문), 030300(저축은행), 050000(보험), 060000(금융투자)
        'topFinGrpNo': '020000',
        'pageNo': 1
    }
    # 응답을 json 형태로 변환
    response = requests.get(url, params=params).json()
    return response



if __name__ == '__main__':
    # json 형태의 데이터 반환
    result = get_deposit_products()
    # prrint.prrint(): json 을 보기 좋은 형식으로 출력
    pprint.pprint(result)
```

이메일로 전달받은 API KEY

세부 요구사항

A. 데이터 추출 - Key 값 출력하기

B. 데이터 추출 - 전체 정기예금 상품 리스트

C. 데이터 가공 - 전체 정기예금 상품들의 옵션 정보 리스트

D. 데이터 가공 - 상품과 옵션 정보들을 담고 있는 새로운 값을 만들어 반환하기

E. 생성형 AI 활용하기

- [참고] 모든 출력 결과는 예시입니다. 날짜에 따라 다르게 나올 수 있습니다.

A. 데이터 추출 - Key 값 출력하기

- 전체 정기예금의 응답을 json 형태로 변환 후 아래와 같이 Key 값만 출력하도록 구성합니다.

```
dict_keys(['prdt_div', 'total_count', 'max_page_no', 'now_page_no', 'err_cd', 'err_msg', 'baseList', 'optionList'])
```

- 공식 문서의 요청 변수 및 예제 요청결과(JSON) 부분을 참고합니다.
- [힌트] 모든 데이터는 JSON 객체의 “result” 키 값으로 조회할 수 있습니다.

B. 데이터 추출 - 전체 정기예금 상품 리스트

- 응답 중 정기예금 상품 리스트 정보만 출력하도록 구성합니다.

- 출력 결과 예시


```
[{'dcls_end_day': None,
  'dcls_month': '202304',
  'dcls_strt_day': '20230517',
  'etc_note': '- 가입기간: 1~36개월\n'
              '- 최소가입금액: 1만원 이상\n'
              '- 만기일을 일,월 단위로 자유롭게 선택 가능\n'
              '- 만기해지 시 신규일 당시 영업점과 인터넷 홈페이지에 고시된 계약기간별 금리 적용',
  'fin_co_no': '0010001',
  'fin_co_subm_day': '202305171024',
  'fin_prdt_cd': 'WR0001B',
  'fin_prdt_nm': 'WON플러스예금',
  'join_deny': '1',
  'join_member': '실명의 개인',
  'join_way': '인터넷,스마트폰,전화(텔레뱅킹)',
  'kor_co_nm': '우리은행',
  'max_limit': None,
  'mtrt_int': '만기 후\n'
              '- 1개월이내 : 만기시점약정이율×50%\n'
              '- 1개월초과 6개월이내: 만기시점약정이율×30%\n'
              '- 6개월초과 : 만기시점약정이율×20%\n'
              '\n'
              '※만기시점 약정이율 : 일반정기예금 금리',
  'spcl_cnd': '해당사항 없음'},
 {'dcls_end_day': '99991231',
  'dcls_month': '202304',
  'dcls_strt_day': '20230420',
```

C. 데이터 가공 - 전체 정기예금 옵션 리스트

- 응답 중 정기예금 상품들의 옵션 리스트를 출력하도록 구성합니다.
- 이 때, 원하는 데이터만 추출하여 출력되는 결과를 아래와 같이 변경하여 반환하는 함수를 작성하시오.
- 출력 결과 예시


```
[{'금융상품코드': 'WR0001B',
  '저축 금리': 3.39,
  '저축 기간': '6',
  '저축금리유형': 'S',
  '저축금리유형명': '단리',
  '최고 우대금리': 3.39},
 {'금융상품코드': 'WR0001B',
  '저축 금리': 3.57,
  '저축 기간': '12',
  '저축금리유형': 'S',
  '저축금리유형명': '단리',
  '최고 우대금리': 3.57},
 {'금융상품코드': 'WR0001B',
  '저축 금리': 3.36,
```


- [참고] Python Dictionary 원하는 키 값으로 데이터 추가하기

```
new_dict = {}
new_dict['추가'] = "test"
print(new_dict)
```

- 출력 결과

```
{'추가': 'test'}
```

D. 데이터 가공 - 새로운 값을 만들어 반환하기(1/2)

- 상품과 옵션 정보들을 담고 있는 새로운 값을 만들어 딕셔너리 형태로 반환하도록 구성합니다.
- 다음과 같은 값만 추출하여 새로운 값에 포함합니다.
  - 금융 상품: '금융회사명', '금융상품명', '금리정보'
  - 해당 금융 상품의 금리 정보(옵션): '저축금리유형', '저축금리유형명', '저축 기간', '저축 금리', '최고 우대금리'
    - 하나의 금융 상품에 대해 여러 개의 옵션을 가질 수 있습니다.
- [힌트] 금융 상품 코드가 같은 금융 상품과 옵션을 하나의 딕셔너리로 만듭니다.

D. 데이터 가공 - 새로운 값을 만들어 반환하기(2/2)

- 출력 결과 예시
  - WON플러스예금에 대한 정보
  - 4가지 금리 정보를 포함하고 있습니다.


```
[{'금리정보': [{'저축 금리': 3.39,
              '저축 기간': '6',
              '저축금리유형': 'S',
              '저축금리유형명': '단리',
              '최고 우대금리': 3.39},
             {'저축 금리': 3.57,
              '저축 기간': '12',
              '저축금리유형': 'S',
              '저축금리유형명': '단리',
              '최고 우대금리': 3.57},
             {'저축 금리': 3.36,
              '저축 기간': '24',
              '저축금리유형': 'S',
              '저축금리유형명': '단리',
              '최고 우대금리': 3.36},
             {'저축 금리': 3.33,
              '저축 기간': '36',
              '저축금리유형': 'S',
              '저축금리유형명': '단리',
              '최고 우대금리': 3.33}],
  '금융상품명': 'WON플러스예금',
  '금융회사명': '우리은행'},
```

E. 생성형 AI 활용하기

- 생성형 AI 를 활용하여 자유롭게 데이터를 수집하고, 추출하고 가공해봅니다.
- 힌트
  - 내가 사용하는 API (금융상품통합비교공시 API)를 잘 알고 있는 지 확인합니다.
    - 질문 추천: OpenWeatherMap api 의 엔드포인트를 알려줘
      - 엔드포인트(Endpoint): 웹 API 에서 특정 기능이나 리소스에 접근할 수 있는 URL
  - 원하는 동작을 자유롭게 정하고, 해당 내용의 정답을 얻을 수 있도록 프롬프트를 구성합니다.
    - 질문 예시: 금융상품통합비교공시 API 를 이용하여 금리가 가장 높은 적금 상품을 가져오고 싶어.
- 프롬프트 화면 캡처, 코드, 코드 실행 결과를 모두 제출합니다.

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

제출

제출 시 주의사항

- 제출기한은 금일 18시까지입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출합니다.
  - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- https://lab.ssafy.com/ 에 프로젝트를 생성하고 제출합니다.
  - 프로젝트 이름은 '프로젝트 번호 + pjt' 로 지정합니다. (ex. 01_pjt)
- 반드시 각 반 담당 교수님을 Maintainer 로 설정해야 합니다.