HW3_201921521_류한별

텍스트 마이닝

텍스트 마이닝이란?

- 일정한 길이의 벡터로 변환하는 것
- 변환된 벡터에 머신러닝 기법을 적용하는 것

텍스트 마이닝 방법

- NLP기본도구
- 머신러닝(딥러닝)

텍스트 마이닝 도구(파이썬)

- NLTK:가장 많이 알려진 NLP라이브러리
- Scikit Learn:머신러닝 라이브러리로 기본적인 NLP, 다양한 텍스트 마이닝 관련 도구를 지원한다.
- Gensim :Word2Vec으로 유명, sklearn과 마찬가지로 다양한 텍스트 관련 도구 지원
- Keras: RNN, seq2seq 등 딥러닝 위주의 라이브러리 제공

텍스트 마이닝 기본도구

- Tokenize:대상이 되는 문서/문장을 최소 단위로 쪼갬
- Text normalization: 최소 단위를 표준화
- POS-tagging: 최소 의미단위로 나누어진 대상에 대해 품사를 부착
- Chunking: POS-tagging의 결과를 명사구, 형용사구, 분사구 등과 같은 말모듬으로 다시 합치는 과정
- BOW, TFIDF: tokenized 결과를 이용하여 문서를 vector로 표현

Tokenize

- Document를 Sentence의 집합으로 분리
- Sentence를 Word의 집합으로 분리
- 의미 없는 문자 등을 걸러 냄

Text normalization

- 동일한 의미의 다른 단어가 다른 형태를 갖는 것을 보완한다.
  - 다른 형태의 단어를 통일시켜 표준 단어로 변환
- 단어의 다양한 변형을 하나로 통일한다.
- 사전을 이용하여 단어의 원형을 추출한다.

POS-tagging

- 토큰화와 정규화를 통해 나누어진 형태소에 대해 품사를 결정하여 할당하는 작업이다.
- 품사를 알기 위해서는 문맥을 파악해야 한다.
- 각 단어에 대해 올바른 발음을 하기 위해 품사태깅을 이용함

- 형태소 분석으로 번역되기도 한다.

Chunking

- 주어와 동사가 없는 두 단어 이상의 집합인 구를 의미한다.
- 주어진 텍스트에서 이와 같은 chunk를 찾는 과정이다.
- 분석결과인 형태소들을 겹치지 않으며 의미있는 구로 묶는 과정
- 정보추출을 하기 위한 전단계 혹은 정보추출에 포함되기도 한다.

Logistic Regression

- 분류를 위한 회귀분석
  - 종속변수와 대립변수간의 관계를 구체적인 함수로 나타내 예측모델에 사용한다.
  - 종속변수가 범주형 데이터를 대상으로 하며 입력 데이터가 주어졌을 때 결과가 나뉘기 때문에 일종의 분류 기법에 해당한다.
- 텍스트 마이닝에서의문제
  - 추정해야 할 계수기 벡터의 크기만큼 존재해 과적합이 발생하기 쉽고 많은 데이터셋이 필요하다.
  - 정규화를 이용해 과적합 해결 노력

Ridge and Lasso Regression

- 릿지 회귀
  - 목적함수에 추정할 계수에 대한 규제항을 추가하여 과적합을 방지
- 라쏘 회귀
  - 남은 단어들로 분류의 이유에 대해 설명이 가능하다는 장점이 있다.

문서분류의 활용

- 네이버 지식백과
  - 소비자의 감성과 관련된 텍스트 정보를 자동으로 추출하는 텍스트 마이닝 기술의 영역

잠재 의미 분석

- 문서 간의 유사도
  - Count vector나 TFIDF에 cosine similarity를 직접 적용하는 경우, 물리적인 단어로만 유사도를 측정하게 된다.
  - 직접적인 단어가 아니라 의미적으로 유사한(문서에서 함께 많이 등장한) 단어들로 유사도를 측정하는 것이 가능할 것으로 기대
- 단어 간의 유사도
  - 주어진 문서 집합에서 단어들이 어떤 유사도를 가지는지 볼 수 있다.

Topic Modeling

- 주제를 의미하는 용어로 사용된다.
- 문서들이 특정한 주제에 속할 확률분포와 주제로부터 특정 단어들이 파생되어 나올 확률분포가 주어졌을 때, 이 두 확률분포를 조합하여 각 문서에 들어가는 단어의 확률분포를 계산한다.
- θ: 문서들이 각 주제들에 속할 확률분포
  - 디리클레분포의 매개변수인 <알파>에 의해 결정

- N: 특정 문서에 속한 단어의 집합
- M: 전체 문서의 집합
- z: 문서 내의 단어들이 주제들에 속할 확률분포
  - θ에 의한 다항분포로 선택
- β: 각 주제가 특정 단어를 생성할 확률을 나타내는 확률분포
- z와 β에 의해 실제 문서들의 단어분포인 w가 결정
- w만이 실제로 문서들을 통해 주어진 분포, 나머지는 모두 잠재변수
- LDA 알고리즘에서는 주어진 문서와 토픽들의 사전확률 분포인 α와 토픽 내에서 단어의 사전확률분포인 β의 파라미터 값을 활용해 반복적인 시뮬레이션을 통해 z와 θ를추정

Word Embedding

- 단어에 대한 벡터의 dimension reduction이 목표
- 단어 표현
  - Term-Document Matrix에서 Document 별 count vector
  - 일반화가 어려움
  - one-hot-encoding: extremely sparse
- one-hot-encoding으로 표현된 단어를 dense vector로 변환
- 변환된 vector를 이용하여 학습
- 최종목적에 맞게 학습에 의해 벡터가 결정된다.
- 학습목적 관점에서의 단어 의미를 내포한다.

One hot encoding

- 각 단어를 모든 문서에서 사용된 단어들의 수 길이의 벡터로 표현 #### Word2Vec
- 문장에 나타난 단어들의 순서를 이용해 word embedding을 수행한다.
- 단어의 위치에 기반하여 의미를 내포하는 벡터를 생성한다.
  - 비슷한 위치에 나타나는 단어들은 비슷한 벡터를 가지게 된다.
  - 단어 간의 유사성을 이용하여 연산이 가능하다. #### ELMo (Embeddings from Language Model)
- 사전 훈련된 언어모델을 사용하는 워드 임베딩 방법론.
- 문맥을 반영하기 위해 개발된 워드 임베딩 기법
- 문맥 파악을 위해 biLSTM으로 학습된 모형을 이용한다.

Document Embedding

- 단어에 대해 dense vector를 생성하지만 dense vector는 여전히 sparse이다.
- 주변 단어들에 더하여 고유한 벡터를 함께 학습함으로써 dense vector를 생성한다.
- dense vector를 이용해 매칭, 분류 등의 작업을 수행한다. #### RBM (Restricted Boltzmann Machine)
- 사전학습을 목적으로 개발
- vanishing gradient 문제 해결을 위해 제안
- 사전학습을 통한 차원 축소에 사용가능

Autoencoder

- RBM과 유사한 개념이며 작동방식은 PCA와 유사하다.

LSTM (Long Short Term Memory)

- RNN의 문제

    - 문장이 길수록 층이 깊은 형태를 갖게 됨- 경사가 소실되는 문제 발생 - 앞부분의 단어 정보가 학습되지 않음
- LSTM: 직통 통로를 만들어 RNN의 문제를 해결

Bi-LSTM

- 단방향 LSTM의 문제
    - 단어 순서가 갖는 문맥정보가 한 방향으로만 학습된다.
    - 자신의 뒤에 오는 단어에 의해 영향을 받는 경우,학습이 되지 않음
- Bi-LSTM
    - 양방향으로 LSTM을 구성하여 두 결과를 합침
    - 양방향 순서를 모두 학습

합성곱 신경망(Convolutional Neural Networks,CNN)

- 이미지 처리를 위해 개발된 신경망으로 인간의 이미지 인식보다 더 나은 인식성능을 보임
- 주변정보를 학습한다는 점에서 자연어 처리에서의 활용분야가 넓어짐
- 합성곱층과 풀링층으로 구성
- 합성곱층은 2차원 이미지에서 특정영역의 특징을 추출하는 역할=연속된 단어의 특징을 추출하는 것과 유사한 특성

Attention

- 출력에 나온 어떤 단어는 입력에 있는 특정 단어들에 민감한 것에 착안
- 입력의 단어들로부터 출력 단어에 직접 링크를 만든다.

웹 크롤링1 - Static Crawling

1. urllib

- 파이썬은 웹 사이트에 있는 데이터를 추출하기 위해 urllib 라이브러리 사용
- 이를 이용해 HTTP 또는 FTP를 사용해 데이터 다운로드 가능
- urllib은 URL을 다루는 모듈을 모아 놓은 패키지
- urllib.request 모듈은 웹 사이트에 있는 데이터에 접근하는 기능 제공, 또한 인증, 리다렉트, 쿠키처럼 인터넷을 이용한 다양한 요청과 처리가 가능

In [1]:
```
from urllib import request
```

1.1. urllib.request를 이용한 다운로드

- urllib.request 모듈에 있는 urlretrieve() 함수 이용
- 다음의 코드는 PNG 파일을 test.png 라는 이름의 파일로 저장하는 예제임

In [2]:
```
# 라이브러리 읽어들이기
from urllib import request

url="http://uta.pw/shodou/img/28/214.png"
savename="test.png"

request.urlretrieve(url, savename)
print("저장되었습니다")
```

```
저장되었습니다
```

1.2. urlopen으로 파일에 저장하는 방법

- request.urlopen()은 메모리에 데이터를 올린 후 파일에 저장하게 된다.

In [3]:
```
# URL과 저장경로 지정하기
url = "http://uta.pw/shodou/img/28/214.png"
savename = "test1.png"
#다운로드
mem = request.urlopen(url).read()
#파일로 저장하기, wb는 쓰기와 바이너리모드
with open(savename, mode="wb") as f:
    f.write(mem)
    print("저장되었습니다..")
```

```
저장되었습니다..
```

1.3. API 사용하기

클라이언트 접속 정보 출력 (기본)

- API는 사용자의 요청에 따라 정보를 반환하는 프로그램
- IP 주소, UserAgent 등 클라이언트 접속정보 출력하는 "IP 확인API" 접근해서 정보를 추출하는 프로그램

In [4]:
```
#데이터 읽어들이기
url="http://api.aoikujira.com/ip/ini"
res=request.urlopen(url)
data=res.read()

#바이너리를 문자열로 변환하기
text=data.decode("utf-8")
print(text)
```

```
[ip]
API_URI=http://api.aoikujira.com/ip/get.php
REMOTE_ADDR=112.168.120.158
REMOTE_HOST=112.168.120.158
REMOTE_PORT=48308
HTTP_HOST=api.aoikujira.com
HTTP_USER_AGENT=Python-urllib/3.8
HTTP_ACCEPT_LANGUAGE=
HTTP_ACCEPT_CHARSET=
SERVER_PORT=80
FORMAT=ini
```

2. BeautifulSoup

- 스크레이핑(Scraping or Crawling)이란 웹 사이트에서 데이터를 추출하고, 원하는 정보를 추출하는 것을 의미
- BeautifulSoup란 파이썬으로 스크레이핑할 때 사용되는 라이브러리로서 HTML/XML에서 정보를 추출할 수 있도록 도와줌. 그러나 다운로드 기능은 없음.
- 파이썬 라이브러리는 pip 명령어를 이용해 설치 가능. Python Package Index(PyPI)에 있는 패키지 명령어를 한줄로 설치 가능

- URL (http://pypi.python.org/pypi)

패키지 import 및 예제 HTML

In [5]:
```
from bs4 import BeautifulSoup
html = """
<html><body>
  <h1>스크레이핑이란?</h1>
  <p>웹 페이지를 분석하는 것</p>
  <p>원하는 부분을 추출하는 것</p>
</body></html>
"""
```

2.1. 기본 사용

- 다음은 Beautifulsoup를 이용하여 웹사이트로부터 HTML을 가져와 문자열로 만들어 이용하는 예제임
- h1 태그를 접근하기 위해 html-body-h1 구조를 사용하여 soup.html.body.h1 이런식으로 이용하게 됨.
- p 태그는 두개가 있어 soup.html.body.p 한 후 next_sibling을 두번 이용하여 다음 p를 추출. 한번만 하면 그 다음 공백이 추출됨.
- HTML 태그가 복잡한 경우 이런 방식으로 계속 진행하기는 적합하지 않음.

2) HTML 분석하기

In [6]:
```
soup = BeautifulSoup(html, 'html.parser')
```

3) 원하는 부분 추출하기

In [7]:
```
h1 = soup.html.body.h1
p1 = soup.html.body.p
p2 = p1.next_sibling.next_sibling
```

4) 요소의 글자 출력하기

In [8]:
```
print(f"h1 = {h1.string}")
print(f"p  = {p1.string}")
print(f"p  = {p2.string}")
```

```
h1 = 스크레이핑이란?
p  = 웹 페이지를 분석하는 것
p  = 원하는 부분을 추출하는 것
```

2.2. 요소를 찾는 method

단일 element 추출: find()

BeautifulSoup는 루트부터 하나하나 요소를 찾는 방법 말고도 find()라는 메소드를 제공함

In [9]:
```
soup = BeautifulSoup(html, 'html.parser')
```

- 1) find() 메서드로 원하는 부분 추출하기

In [10]:
```
title = soup.find("h1")
body  = soup.find("p")
print(title)
```

```
<h1>스크레이핑이란?</h1>
```

- 2) 텍스트 부분 출력하기

In [11]:
```
print(f"#title = {title.string}" )
print(f"#body = {body.string}")
```

```
#title = 스크레이핑이란?
#body = 웹 페이지를 분석하는 것
```

복수 elements 추출: find_all()

여러개의 태그를 한번에 추출하고자 할때 사용함. 다음의 예제에서는 여러개의 태그를 추출하는 법을 보여주고 있음

In [12]:
```
html = """
<html><body>
  <ul>
    <li><a href="http://www.naver.com">naver</a></li>
    <li><a href="http://www.daum.net">daum</a></li>
  </ul>
</body></html>
"""

soup = BeautifulSoup(html, 'html.parser')
```

- 1) find_all() 메서드로 추출하기

In [13]:
```
links = soup.find_all("a")
print(links, len(links))
```

```
[<a href="http://www.naver.com">naver</a>, <a href="http://www.daum.net">daum</a>] 2
```

- 2) 링크 목록 출력하기

In [14]:
```
for a in links:
    href = a.attrs['href'] # href의 속성에 있는 속성값을 추출
    text = a.string
    print(text, ">", href)
```

```
naver > http://www.naver.com
daum > http://www.daum.net
```

3. Css Selector

Css Selector란, 웹상의 요소에 css를 적용하기 위한 문법으로, 즉 요소를 선택하기 위한 패턴입니다.

출처: https://www.w3schools.com/cssref/css_selectors.asp

앞서 간단하게 태그를 사용하여 데이터를 추출하는 방법에 대해서 살펴보았습니다.

하지만 복잡하게 구조화된 웹 사이트에서 자신이 원하는 데이터를 가져오기 위해서는 Css Selector에 대한 이해가 필요합니다.

BeautifulSoup에서 Css Selector 사용하기

BeautifulSoup에서는 Css Selector로 값을 가져올 수 있도록 find와는 다른 다음과 같은 메서드를 제공합니다.

In [15]:
```
html = """
<html><body>
<div id="meigen">
  <h1>위키북스 도서</h1>
  <ul class="items">
    <li>유니티 게임 이펙트 입문</li>
    <li>스위프트로 시작하는 아이폰 앱 개발 교과서</li>
    <li>모던 웹사이트 디자인의 정석</li>
  </ul>
</div>
</body></html>
"""

# HTML 분석하기
soup = BeautifulSoup(html, 'html.parser')
```

- 필요한 부분을 CSS 쿼리로 추출하기

In [16]:
```
# 타이틀 부분 추출하기 --- (※3)
h1 = soup.select_one("div#meigen > h1").string
print(f"h1 = {h1}")

# 목록 부분 추출하기 --- (※4)
li_list = soup.select("div#meigen > ul.items > li")
for li in li_list:
  print(f"li = {li.string}")
```

```
h1 = 위키북스 도서
li = 유니티 게임 이펙트 입문
li = 스위프트로 시작하는 아이폰 앱 개발 교과서
li = 모던 웹사이트 디자인의 정석
```

4. 활용 예제

앞서 배운 urllib과 BeautifulSoup를 조합하면, 웹스크레이핑 및 API 요청 작업을 쉽게 수행하실 수 있습니다.

1. URL을 이용하여 웹으로부터 html을 읽어들임 (urllib)
2. html 분석 및 원하는 데이터를 추출 (BeautifulSoup)

In [18]:
```
from bs4 import BeautifulSoup
from urllib import request, parse
```

4.1. 네이버 금융 - 환율 정보

- 다양한 금융 정보가 공개돼 있는 "네이버 금융"에서 원/달러 환율 정보를 추출해보자!
- 네이버 금융의 시장 지표 페이지 https://finance.naver.com/marketindex/
- 다음은 원/달러 환율 정보를 추출하는 프로그램임

1) HTML 가져오기

In [19]:
```
url = "https://finance.naver.com/marketindex/"
res = request.urlopen(url)
```

2) HTML 분석하기

In [20]:
```
soup = BeautifulSoup(res, "html.parser")
```

3) 원하는 데이터 추출하기

In [21]:
```
price = soup.select_one("div.head_info > span.value").string
print("usd/krw =", price)
```

```
usd/krw = 1,178.00
```

4.2. 기상청 RSS

- 기상청 RSS에서 특정 내용을 추출하는 예제
- 기상청 RSS에서 XML 데이터를 추출하고 XML 내용을 출력
- 기상청의 RSS 서비스에 지역 번호를 지정하여 데이터 요청해보기
  http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp
  - 참고: 기상청 RSS http://www.kma.go.kr/weather/lifenindustry/service_rss.jsp
- 파이썬으로 요청 전용 매개변수를 만들 때는 urllib.parse 모듈의 urlencode() 함수를 사용해 매개변수를 URL로 인코딩한다.

1) HTML 가져오기

In [22]:
```
url = "http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"

#매개변수를 URL로 인코딩한다.
values = {
    'stnId':'109'
}

params=parse.urlencode(values)
url += "?"+params # URL에 매개변수 추가
print("url=", url)

res = request.urlopen(url)
```

```
url= http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=109
```

2) HTML 분석하기

In [23]:
```
soup = BeautifulSoup(res, "html.parser")
```

3) 원하는 데이터 추출하기

In [24]:
```
header = soup.find("header")

title = header.find("title").text
wf = header.find("wf").text

print(title)
print(wf)
```

```
서울,경기도 육상중기예보
○ (강수) 29일(수)은 비가 내리겠습니다.<br />○ (기온) 이번 예보기간 아침최저기온은 13~20도로 어제(24일, 아침최저기온 13~18도)와 비슷하거나 조금 높겠고, <br />          낮최고기온은 23~28도로 어제(24일, 낮최고기온 26~27도)와 비슷하겠습니다.<br />○ (해상) 서해중부해상의 물결은 0.5~2.0m로 일겠습니다.
```

- css selector 기반

In [25]:
```
title = soup.select_one("header > title").text
wf = header.select_one("header wf").text

print(title)
print(wf)
```

```
서울,경기도 육상중기예보
○ (강수) 29일(수)은 비가 내리겠습니다.<br />○ (기온) 이번 예보기간 아침최저기온은 13~20도로 어제(24일, 아침최저기온 13~18도)와 비슷하거나 조금 높겠고, <br />          낮최고기온은 23~28도로 어제(24일, 낮최고기온 26~27도)와 비슷하겠습니다.<br />○ (해상) 서해중부해상의 물결은 0.5~2.0m로 일겠습니다.
```

4.3. 윤동주 작가의 작품 목록

- 위키문헌 (https://ko.wikisource.org/wiki) 에 공개되어 있는 윤동주의 작품목록을 가져오기
- 윤동주 위키 (https://ko.wikisource.org/wiki/%EC%A0%80%EC%9E%90:%EC%9C%A4%EB%8F%99%EC%A3%E
- 하늘과 바람과 시 부분을 선택한 후 오른쪽 마우스 이용해 copy selector로 카피하면 다음의 CSS 선택자가 카피됨
  - #mw-content-text > div > ul:nth-child(6) > li > b > a
- nth-child(n) 은 n 번째 요소를 의미 즉 6번째 요소를 의미, #mw-content-text 내부에 있는 url 태그는 모두 작품과 관련된 태그. 따라서 따로 구분할 필요는 없으며 생략해도 됨. BeautifulSoup는 nth-child 지원하지 않음
  - Recall PR7 Problem1

In [26]:
```
#뒤의 인코딩 부분은 "저자:윤동주"라는 의미입니다.
# 따로 입력하지 말고 위키 문헌 홈페이지에 들어간 뒤에 주소를 복사해서 사용하세요.

url = "https://ko.wikisource.org/wiki/%EC%A0%80%EC%9E%90:%EC%9C%A4%EB%8F%99%EC%A3%BC"
res = request.urlopen(url)
soup = BeautifulSoup(res, "html.parser")

# #mw-content-text 바로 아래에 있는
# ul 태그 바로 아래에 있는
# li 태그 아래에 있는
# a 태그를 모두 선택합니다.
a_list = soup.select("#mw-content-text   ul > li  a")
for a in a_list:
    name = a.string
    print(f"- {name}", )
```

- 하늘과 바람과 별과 시
- 증보판
- 서시
- 자화상
- 소년
- 눈 오는 지도
- 돌아와 보는 밤
- 병원
- 새로운 길
- 간판 없는 거리
- 태초의 아침
- 또 태초의 아침
- 새벽이 올 때까지
- 무서운 시간
- 십자가
- 바람이 불어
- 슬픈 족속
- 눈감고 간다
- 또 다른 고향
- 길
- 별 헤는 밤
- 흰 그림자
- 사랑스런 추억
- 흐르는 거리
- 쉽게 씌어진 시
- 봄
- 참회록
- 간(肝)
- 위로
- 팔복
- 못자는밤
- 달같이
- 고추밭
- 아우의 인상화
- 사랑의 전당
- 이적
- 비오는 밤
- 산골물
- 유언
- 창
- 바다
- 비로봉
- 산협의 오후
- 명상
- 소낙비
- 한난계
- 풍경
- 달밤
- 장
- 밤
- 황혼이 바다가 되어
- 아침
- 빨래
- 꿈은 깨어지고
- 산림
- 이런날
- 산상
- 양지쪽
- 닭
- 가슴 1
- 가슴 2
- 비둘기
- 황혼
- 남쪽 하늘
- 창공
- 거리에서
- 삶과 죽음
- 초한대
- 산울림

- 해바라기 얼굴
- 귀뚜라미와 나와
- 애기의 새벽
- 햇빛 · 바람
- 반디불
- 둘 다
- 거짓부리
- 눈
- 참새
- 버선본
- 편지
- 봄
- 무얼 먹구 사나
- 굴뚝
- 햇비
- 빗자루
- 기왓장 내외
- 오줌싸개 지도
- 병아리
- 조개껍질
- 겨울
- 트루게네프의 언덕
- 달을 쏘다
- 별똥 떨어진 데
- 화원에 꽃이 핀다
- 종시

일반문제

In [27]:
```
from bs4 import BeautifulSoup
from urllib import request
```

1. 네이버 뉴스 헤드라인

배운 내용을 바탕으로 네이버 뉴스(https://news.naver.com/)에서 헤드라인 뉴스의 제목을 추출해 보고자 합니다.

Q: 다음의 코드에 css selector를 추가하여 최신 기사의 헤드라인을 스크레이핑하는 코드를 완성하시오.

In [28]:
```
url = "https://news.naver.com/" #네이버 뉴스 주소

res = request.urlopen(url)
soup = BeautifulSoup(res, "html.parser")

selector = "#today_main_news > div.hdline_news > ul > li > div.hdline_article_tit > a"

for a in soup.select(selector):#최신 기사 헤드라인 스크래핑
    title = a.text
    print(title)
```

‘오징어 게임’ 탓 장난전화 ‘폭탄’…“징역형 처벌도 가능”[촉!]

실명계좌 무산된 거래소 코인, 뺄까 옮길까 · · · 투자자 혼란[발칙한 금융]

北, 종전선언에 ‘미군철수’ 조건 걸어… 文은 “한미동맹과 무관”

쿼드 첫 대면 정상회담…중국 겨냥 "자유 · 개방 인도태평양" 역설

中 "모든 코인 거래 불법"...가상화폐 급락

2. 시민의 소리 게시판

다음은 서울시 대공원의 시민의 소리 게시판 입니다.

https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgList.do?pgno=1

해당 페이지에 나타난 게시글들의 제목을 수집하고자 합니다.

Q: 다음의 코드에 css selector를 추가하여 해당 페이지에서 게시글의 제목을 스크레이핑하는 코드를 완성하시오. 또한 과제 제출시 하단의 추가 내용을 참고하여 수집한 데이터를 csv 형태로 저장하여 해당 csv 파일도 함께 제출하시오.

In [29]:
```
url_head = "https://www.sisul.or.kr"

url_board = url_head + "/open_content/childrenpark/qna/qnaMsgList.do?pgno=1"



res = request.urlopen(url_board)
soup = BeautifulSoup(res, "html.parser")

# selector = "#detail_con > div.generalboard > table > tbody > tr > td.left.title > a"
selector = "#detail_con > div.generalboard > table > tbody > tr > td.left.title > a"
titles = []
links = []
for a in soup.select(selector):
    titles.append(a.text)
    links.append(url_head + a.attrs["href"])

print(titles, links)
```

```
['어린이를 위한 공원내 식당에 아기를 위한 시설 부족(아기의자가 왜 없죠?)', '강창수 해설사님 ', '동물해설사님 칭찬', '강창수 동물 해설사님', '놀이동산 푸드코트 김치가 중국산인 이유는?', '주슨트 설명 최고예요!!', '강창수 주슨트님 최고 !!', 'ZOOCENT 스케줄표?', '호주동물 호주설명 ', '호주및 호주동물 설명에 대해 '] ['https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210923000005&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210920000001&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210919000004&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210919000003&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210918000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210909000001&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210908000004&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhluRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210906000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpa
```

```
rk/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhIuRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210904000006&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=GJvvAif7wZ34gwWxdwcSCEbWS7L4SLa6HV8UhIuRyrFvL8TKp0CCTLPyC9NpBfl5.etisw2_servlet_user?qnaid=QNAS20210904000004&pgno=1']
```

In [ ]:

추가 내용

수집된 자료를 데이터프레임으로 만들어 csv로 저장하는 것이 일반적입니다

In [30]:

```
import pandas as pd


board_df = pd.DataFrame({"title": titles, "link": links})
board_df.head()
```

Out[30]:

	title	link
0	어린이를 위한 공원내 식당에 아기를 위한 시설 부족(아기의자가 왜 없죠?)	https://www.sisul.or.kr/open_content/childrenp...
1	강창수 해설사님	https://www.sisul.or.kr/open_content/childrenp...
2	동물해설사님 칭찬	https://www.sisul.or.kr/open_content/childrenp...
3	강창수 동물 해설사님	https://www.sisul.or.kr/open_content/childrenp...
4	놀이동산 푸드코트 김치가 중국산인 이유는?	https://www.sisul.or.kr/open_content/childrenp...

In [31]:

```
board_df.to_csv("board.csv", index=False)
```

웹 크롤링2 - Dynamic Crawling

0. 라이브러리

In [114]:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from bs4 import BeautifulSoup
import pandas as pd
from pandas import DataFrame
import time
```

1. Selenium 기초

자신의 크롬 버전을 확인하고 크롬 웹드라이버를 다운받아놓아야합니다.

- 2020.09.13 기준 최신 버전: 85.0.4183.102 1.1. Simple Text Crawling 멜론 사이트에서 노래 제목을 크롤링해보자

URL: https://www.melon.com/chart/index.htm

In [132]:
```
DRIVER_PATH = 'C:/Users/gksquf/OneDrive/Documents/CDesing/chromedriver'
```

In [133]:
```
# chrome driver 설정
driver = webdriver.Chrome(DRIVER_PATH)
driver.implicitly_wait(10)

url = "https://www.melon.com/chart/index.htm"

driver.get(url)
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

# title crawling
title = WebDriverWait(driver, 20) W
    .until(EC.presence_of_element_located((By.CSS_SELECTOR, "#frm > div > table > tbod

# print("Title: {}".format(title.text))

title.text
```

Out[133]: 'STAYWnThe Kid LAROI, Justin Bieber'

css selector의 규칙을 찾아본다

- 1번째 제목: #frm > div > table > tbody > tr:nth-child(1) > td:nth-child(4) > div > div"
- 2번째 제목: #frm > div > table > tbody > tr:nth-child(2) > td:nth-child(4) > div > div

...

- 100번째 제목: #frm > div > table > tbody > tr:nth-child(100) > td:nth-child(4) > div > div 또는 XPATH로도 확인해보자 (full Xpath)
- 1번째 제목: //*[@id="frm"]/div/table/tbody/tr[1]/td[4]/div/div
- 2번째 제목: //*[@id="frm"]/div/table/tbody/tr[2]/td[4]/div/div

...

- 50번째 제목: //*[@id="frm"]/div/table/tbody/tr[100]/td[4]/div/div

In [117]:
```
# 2번째 제목 크롤링
WebDriverWait(driver, 20) W
    .until(EC.presence_of_element_located((By.XPATH, "//*[@id='frm']/div/table/tbody/
```

Out[117]: 'My UniverseWnColdplay, 방탄소년단'

1.2. Text Crawling with for loop

위에서 찾은 Xpath의 규칙을 바탕으로 for loop 만들자

In [118]:
```
 # chrome driver 설정
driver = webdriver.Chrome(DRIVER_PATH)
driver.implicitly_wait(10)
```

```
url = "https://www.melon.com/chart/index.htm"

driver.get(url)
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

# 빈 리스트 변수
title_list = []

# title crawling (TOP 50)
for i in range(1, 51):
    title = WebDriverWait(driver, 20) W
        .until(EC.presence_of_element_located((By.XPATH, f"//*[@id='frm']/div/table/t
    title_list.append(title.text)

print(title_list)
```

```
['STAYWnThe Kid LAROI, Justin Bieber', 'My UniverseWnColdplay, 방탄소년단', '신호등Wn
이무진', 'Permission to DanceWn방탄소년단', 'OHAYO MY NIGHTWn디핵 (D-Hack), PATEKO (파
테코)', 'Next LevelWnaespa', 'ButterWn방탄소년단', '바라만 본다WnMSG워너비(M.O.M)',
'낙하 (with 아이유)WnAKMU (악뮤)', 'WeekendWn태연 (TAEYEON)', 'DynamiteWn방탄소년단',
'좋아좋아Wn조정석', 'QueendomWnRed Velvet (레드벨벳)', 'Peaches (Feat. Daniel Caesar &
Giveon)WnJustin Bieber', 'DUMB DUMBWn전소미', 'Bad HabitsWnEd Sheeran', '시간을 거슬러
(낮에 뜨는 달 X 케이윌)Wn케이윌', '다정히 내 이름을 부르면Wn경서예지, 전건호', '이제
나만 믿어요Wn임영웅', '헤픈 우연Wn헤이즈 (Heize)', '가을 타나 봐Wn이무진', 'StickerWnN
CT 127', '비와 당신Wn이무진', '별빛 같은 나의 사랑아Wn임영웅', 'Savage Love (Laxed - S
iren Beat) (BTS Remix)WnJawsh 685, Jason Derulo, 방탄소년단', "롤린 (Rollin')Wn브레이
브걸스", 'Dun Dun DanceWn오마이걸 (OH MY GIRL)', '작은 것들을 위한 시 (Boy With Luv)
(Feat. Halsey)Wn방탄소년단', '라일락Wn아이유', '그대라는 사치Wn임영웅', '고백Wn멜로망
스', 'HEROWn임영웅', 'ASAPWnSTAYC(스테이씨)', 'LemonadeWnNCT 127', '다시 사랑한다면
(김필 Ver.)Wn임영웅', 'CelebrityWn아이유', '봄날Wn방탄소년단', '사이렌 Remix (Feat. UN
EDUCATED KID, Paul Blanco)Wn호미들', '색안경 (STEREOTYPE)WnSTAYC(스테이씨)', '비가 오
는 날엔 (2021)Wn헤이즈 (Heize)', '끝사랑Wn임영웅', 'Life Goes OnWn방탄소년단', '가을
우체국 앞에서Wn김대명', 'Bk LoveWn임영웅', '잊었니Wn임영웅', '찰나가 영원이 될 때 (The
Eternal Moment)Wn마크툽 (MAKTUB)', '밝게 빛나는 별이 되어 비춰줄게Wn송이한', '하루만
더Wn빅마마', 'Road TripWnNCT 127', '내 손을 잡아Wn아이유']
```

1.3. Text Crawling (Click & Back)

클릭하고 나오기 -> 동적 크롤링 가능 (가사 크롤링 가능)

노래 제목에 링크가 걸려있기 때문에, 해당 링크까지의 XPath를 추가한다.

In [127]:

```
 # chrome driver 설정
driver = webdriver.Chrome(DRIVER_PATH)
driver.implicitly_wait(10)

url = "https://www.melon.com/chart/index.htm"

driver.get(url)
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

# 1번째 click하기
click_element = WebDriverWait(driver, 20).until(EC.presence_of_element_located((By.XP
click_element.click()

# back
driver.back()


# 2번째 click하기
click_element = WebDriverWait(driver, 20).until(EC.presence_of_element_located((By.XP
click_element.click()
```

```
# back
driver.back()
```

1.4. Text Crawling including contents

- 1.2처럼 for문과 함께 써보자! (첫 페이지 5개의 글에 대해 title, artist, heart(하트 갯수), lyrics(가사)를 크롤링
- 1.3에서 사용한 click & back을 활용하자

In [126]:

```
 # chrome driver 설정
driver = webdriver.Chrome(DRIVER_PATH)
driver.implicitly_wait(10)

url = "https://www.melon.com/chart/index.htm"
driver.get(url)
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

# 빈 리스트 변수
title_list = []
artist_list = []
heart_list = []
lyrics_list = []

# crawling (TOP 5) 5개씩 크롤링하기
for i in range(1, 6):
    # click
    click_element = WebDriverWait(driver, 20) ₩
        .until(EC.presence_of_element_located((By.XPATH, f'//*[@id="frm"]/div/table/t
    click_element.click()

    # title crawling 제목
    title = WebDriverWait(driver, 20).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "#downloadfrm > div > div > 
    title_list.append(title.text)

    # artist crawling 가수
    artist = WebDriverWait(driver, 20).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "#downloadfrm > div > div > 
    artist_list.append(artist.text)

    # heart crawling 좋아요
    heart = WebDriverWait(driver, 20).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "#d_like_count")))
    heart_list.append(heart.text)

    # lyrics crawling가사
    lyrics = WebDriverWait(driver, 20).until(
        EC.presence_of_element_located((By.CSS_SELECTOR, "#d_video_summary")))
    lyrics_list.append(lyrics.text)

    # back
    driver.back()

print(title_list)
print(artist_list)
print(heart_list)
print(lyrics_list)
```

['STAY', 'My Universe', '신호등', 'Permission to Dance', 'OHAYO MY NIGHT']
['The Kid LAROI', 'Coldplay', '이무진', '방탄소년단', '디핵 (D-Hack)']
['151,352', '62,896', '214,613', '175,970', '132,687']
["I do the same thing I told youWnthat I never wouldWnI told you I'd changeWneven when I knew I never couldWnI know that I can't findWnnobody elseWnas good as youWnI need you to stayWnneed you to stay hey OhWnI get drunk wake upWnI'm wasted stillWnI realize the timeWnthat I wasted hereWnI feel like you can'tWnfeel the way I feelWnOh I'll be fucked upWnif you can't be right hereWnOh ooh-woahWnOh ooh-woah ooh-woahWnOh ooh-woahWnOh ooh-woah ooh-woahWnOh ooh-woahWnOh ooh-woah ooh-woahWnOh I'll be fucked upWnif you can't be right hereWnI do the same thing I told youWnthat I never wouldWnI told you I'd changeWneven when I knew I never couldWnI know that I can't findWnnobody elseWnas good as youWnI need you to stayWnneed you to stay heyWnI do the same thing I told youWnthat I never wouldWnI told you I'd changeWneven when I knew I never couldWnI know that I can't findWnnobody elseWnas good as youWnI need you to stayWnneed you to stay heyWnWhen I'm away from youWnI miss your touch OohWnYou're the reasonWnI believe in love OohWnIt's been difficultWnfor me to trust OohWnAnd I'm afraidWnthat I'ma fuck it up OohWnAin't no wayWnthat I can leave you strandedWn'Cause you ain't ever left meWnempty-handedWnAnd you know that I knowWnthat I can't live without youWnSo baby stayWnOh ooh-woahWnOh ooh-woah ooh-woahWnOh ooh-woahWnOh ooh-woah ooh-woahWnOh ooh-woahWnOh ooh-woah ooh-woahWnI'll be fucked upWnif you can't be right hereWnI do the same thing I told youWnthat I never wouldWnI told you I'd changeWneven when I knew I never couldWnI know that I can't findWnnobody elseWnas good as youWnI need you to stayWnneed you to stay heyWnI do the same thingWnI told you that I never wouldWnI told you I'd changeWneven when I knew I never couldWnI know that I can't findWnnobody elseWnas good as youWnI need you to stayWnneed you to stay heyWnWoah-ohWnI need you to stayWnneed you to stay hey", 'You, you are my universe andWnI just want to put you firstWnAnd you, you are my universe, and I…WnWnIn the night I lie and look up at youWnWhen the morning comes I watch you riseWnThere's a paradise they couldn't captureWnThat bright infinity inside your eyesWnWn매일 밤 네게 날아가 (가)Wn꿈이란 것도 잊은 채Wn나 웃으며 너를 만나 (나)WnNever ending forever babyWnWnYou, you are my universe andWnI just want to put you firstWnAnd you, you are my universe, andWnYou make my world light up insideWnWn어둠이 내겐 더 편했었지Wn길어진 그림자 속에서 (eyes)WnWnAnd they said that we can't be togetherWnBecauseWnBecause we come from different sidesWnWnYou, you are my universe andWnI just want to put you firstWnAnd you, you are my universe, andWnYou make my world light up insideWnWnMy universe (do do, do do)WnMy universe (do do, do do)WnMy universe (do do, do do)Wn(you make my world)WnYou make my world light up insideWnWnMake my world light up insideWnWn나를 밝혀주는 건Wn너란 사랑으로 수 놓아진 별Wn내 우주의 넌Wn또 다른 세상을 만들어 주는 걸WnWn너는 내 별이자 나의 우주니까Wn지금 이 시련도 결국엔 잠시니까Wn너는 언제까지나 지금처럼 밝게만 빛나줘Wn우리는 너를 따라 이 긴 밤을 수놓을 거야WnWn너와 함께 날아가 (가)WnWhen I'm without you I'm crazyWn자 어서 내 손을 잡아 (아)WnWe are made of each other babyWnWnYou, you are my universe andWnI just want to put you firstWnAnd you, you are my universe, andWnYou make my world light up insideWnWnMy universe (you, you are)WnMy universe (I just want)WnMy universe (you, you are)WnMy universe, and IWnWnMy universe', '이제야 목적지를 정했지만Wn가려한 날 막아서네 난 갈 길이 먼데Wn새빨간 얼굴로 화를 냈던Wn친구가 생각나네Wn이미 난 발걸음을 떼었지만Wn가려한 날 재촉하네 걷기도 힘든데Wn새파랗게 겁에 질려 도망간Wn친구가 뇌에 맴도네Wn건반처럼 생긴 도로 위Wn수많은 동그라미들 모두가Wn멈췄다 굴렀다 말은 잘 들어Wn그건 나도 문제가 아냐Wn붉은색 푸른색 그 사이Wn3초 그 짧은 시간Wn노란색 빛을 내는Wn저기 저 신호등이Wn내 머릿속을 텅 비워버려Wn내가 빠른지도Wn느린지도 모르겠어Wn그저 눈앞이 샛노랄 뿐이야Wn솔직히 말하자면 차라리Wn운전대를 못 잡던 어릴 때가Wn더 좋았었던 것 같아Wn그땐 함께 온 세상을Wn거닐 친구가 있었으니Wn건반처럼 생긴 도로 위Wn수많은 조명들이 날 빠르게Wn번갈아 가며 비추고 있지만Wn난 아직 초짜란 말이야Wn붉은색 푸른색 그 사이Wn3초 그 짧은 시간Wn노란색 빛을 내는 저기 저 신호등이Wn내 머릿속을 텅 비워버려Wn내가 빠른지도Wn느린지도 모르겠어Wn그저 눈앞이 샛노랄 뿐이야Wn꼬질꼬질한 사람이나Wn부자 곁엔 아무도 없는Wn삼색 조명과 이색 칠 위에Wn서 있어 괴롭히지 마Wn붉은색 푸른색 그 사이Wn3초 그 짧은 시간Wn노란색 빛을 내는 저기 저 신호등이Wn내 머릿속을 텅 비워버려Wn내가 빠른지도Wn느린지도 모르겠어Wn그저 눈앞이 샛노랄 뿐이야', 'It's the thought of being youngWnWhen your heart's just like a drumWnBeating louder with no way to guard itWnWhen it all seems like it's wrongWnJust sing along to Elton JohnWnAnd to that feeling, we're just getting startedWnWnWhen the nights get colderWnAnd the rhythms got you falling behindWnJust dream about that momentWnWhen you look yourself right in the eye, eye, eyeWnThen you sayWnWnI wanna danceWnThe music's got me goingWnAin't nothing that can stop how we move yeahWnLet's break our plansWnAnd live just like we're goldenWnAnd roll in like we're dancing foolsWnWnWe don't need to worryWn'Cause when we fall we know how to landWnDon't need to talk the talk, just walk the walk tonightWn'Cause we don't need permission to danceWnWnThere's always something that's standing in the wayWnBut if you don't let it faze yaWnYou'll know just how to breakWnJust keep the right vibe yeahWn'Cause there's no looking backWnThere ain't no one to proveWnWe don't got this on lock yeahWnWnThe wait is overWnThe time is now so let's do it rightWnYeah we'll keep goingWnAnd stay up until we see th

e sunriseWnAnd we' ll sayWnWnI wanna danceWnThe music' s got me goingWnAin' t nothing that can stop how we move yeahWnLet' s break our plansWnAnd live just like we' re goldenWnAnd roll in like we' re dancing foolsWnWnWe don' t need to worryWn 'Cause when we fall we know how to landWnDon' t need to talk the talk, just walk the walk tonight Wn 'Cause we don' t need permission to danceWnWnDa na na na na na naWnDa na na na na n a naWnDa na na na na na naWnNo, we don' t need permission to danceWnWnDa na na na na n a naWnDa na na na na na naWnDa na na na na na naWnWnWell let me show yaWnThat we can k eep the fire aliveWn 'Cause it' s not overWnTill it' s over say it one more timeWnSay WnWnI wanna danceWnThe music' s got me goingWnAin' t nothing that can stop how we move yeahWnLet' s break our plansWnAnd live just like we' re goldenWnAnd roll in like we' r e dancing foolsWnWnWe don' t need to worryWn 'Cause when we fall we know how to landWn Don' t need to talk the talk, just walk the walk tonightWn 'Cause we don' t need permi ssion to dance', '너를 사랑하고 있어Wn너를 사랑하고 있어Wn자기야 날 사랑해주면 안 될까 Wn말처럼 쉽지는 않은 걸 알지만Wn세게 날 안아주면 안 될까Wn오늘따라 세상이 무섭단 말이 야Wn잠깐 인공호흡을 해주라Wn왠지 숨이 잘 안 쉬어져서 난Wn날 놓을 거면 과거에 놔주라Wn 네가 있는 시간에서 죽어갈 거야Wn우리 그냥 결혼하면 안 될까Wn돈은 내가 열심히 벌 테니까 Wn이 세상과 내가 눈감는 날Wn까지만 날 사랑한다 말해주라Wn내가 너를 사랑해도Wn네가 날 안 사랑해도Wn우린 나름대로 행복할 거야Wn내 방 천장에 그려 본Wn내 우주에게 물어본Wn말은 나를 사랑하면 안 될까Wn오사카나 오키나와의 바다Wn내 뮤비들을 찍었던 곳 말이야Wn같이 가 자 약속했었잖아Wn그 약속이 깨질까 봐 겁이 나WnWHUTUF이 결혼한다 하던 날Wn진짜 처음으로 걔가 부럽더라Wn하얀 웨딩드레스를 입은 아름다운Wn너와 영원을 말할 수 있을까Wn가족이 되 어주라Wn내 집이 되어주라Wn나도 날 줄 테니 너도 널 주라Wn평생의 연인이야Wn네 말대로 말 이야Wn그래 별과 우주잖아Wn날 사랑하지 않는다면Wn나의 사랑 반을 받아Wn남은 사랑의 반도 Wn내가 채워줄 거야 꼭Wn내가 너를 사랑해도Wn네가 날 안 사랑해도Wn우린 나름대로 행복할 거야Wn내 방 천장에 그려 본Wn내 우주에게 물어본Wn말은 나를 사랑하면 안 될까Wn내가 너를 사랑해도Wn네가 날 안 사랑해도Wn우린 나름대로 행복할 거야Wn내 방 천장에 그려 본Wn내 우 주에게 물어본Wn말은 나를 사랑하면 안 될까']

TIP: 보통은 결과값을 데이터프레임 형태로 저장한다

In [128]:
```
# 결과 변수
raw_result = {'title': title_list,
              'artist': artist_list,
              'heart': heart_list,
           'lyrics': lyrics_list}

result = pd.DataFrame(raw_result)

# # csv 파일로 save
# result.to_csv("MelonTop5", mode='w')

# driver 종료
driver.quit()
```

In [129]:
```
result
```

Out[129]:

	title	artist	heart	lyrics
0	STAY	The Kid LAROI	151,352	I do the same thing I told you\nthat I never w...
1	My Universe	Coldplay	62,896	You, you are my universe and\nI just want to p...
2	신호등	이무진	214,613	이제야 목적지를 정했지만\n가려한 날 막아서네 난 갈 길이 먼데 \n새빨간 얼굴로 화...
3	Permission to Dance	방탄소년단	175,970	It's the thought of being young\nWhen your hea...
4	OHAYO MY NIGHT	디핵 (D-Hack)	132,687	너를 사랑하고 있어\n너를 사랑하고 있어\n자기야 날 사랑해주면 안 될까\n말처럼 ...

2. Image Crawling

이미지 크롤링하기

- 1번째 이미지:
  /html/body/div/div[3]/div/div/div[4]/form/div/table/tbody/tr[1]/td[4]/div/a/img
- 2번째 이미지:
  /html/body/div/div[3]/div/div/div[4]/form/div/table/tbody/tr[2]/td[4]/div/a/img

...

- 50번째 이미지:
  /html/body/div/div[3]/div/div/div[4]/form/div/table/tbody/tr[50]/td[4]/div/a/img ####
  STEP1. URL Crawling

In [130]:
```
 # chrome driver 설정
driver = webdriver.Chrome(DRIVER_PATH)
driver.implicitly_wait(10)

url = "https://www.melon.com/chart/index.htm"

driver.get(url)
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')

# 빈 리스트 변수
link_list = []

# # img crawling (TOP 50)
for i in range(1, 51):# 50개의 이미지 크롤링하기

    img = WebDriverWait(driver, 20) W
        .until(EC.presence_of_element_located((By.CSS_SELECTOR, f"#frm > div > table 

    link_list.append(img.get_attribute('src'))#빈 리스트에 추가하여 저장

print(link_list)
```

['https://cdnimg.melon.co.kr/cm2/album/images/106/46/395/10646395_20210707141710_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/107/20/913/10720913_20210923173742_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/07/796/10607796_20210513201807_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/48/182/10648182_20210709104950_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/104/47/520/10447520_20200619123343_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/09/232/10609232_20210517155130_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/12/483/10612483_20210521111412_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/38/275/10638275_20210625172521_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/61/658/10661658_20210726111159_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/45/654/10645654_20210706155154_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/104/79/150/10479150_20200918102847_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/53/694/10653694_20210715164901_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/80/450/10680450_20210813124748_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/80/103/10580103_20210319132819_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/67/450/10667450_20210802111127_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/37/411/10637411_20210909170255_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/98/116/10698116_20210831104635_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/10/525/10610525_20210518143433_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/c

m2/album/images/104/12/319/10412319_20200403103006_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/11/845/10611845_20210520170350_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/107/12/767/10712767_20210913165623_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/107/03/942/10703942_20210917110116_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/31/122/10631122_20210617142653_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/75/005/10575005_20210309113840_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/104/98/123/10498123_20201002094556_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/100/43/575/10043575_20210302112520_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/04/729/10604729_20210510143932_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm/album/images/102/73/641/10273641_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/54/246/10554246_20210325161233_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/44/845/10644845_20210705203115_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/95/590/10695590_20210827162225_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/13/079/10513079_20201103201136_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/89/127/10589127_20210407175809_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/107/03/942/10703942_20210917110116_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/14/238/10614238_20210525100205_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/54/246/10554246_20210325161233_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm/album/images/100/37/969/10037969_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/80/227/10580227_20210319163608_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/107/04/178/10704178_20210906141809_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/70/618/10670618_20210804111639_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/39/384/10639384_20210628195604_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/21/521/10521521_20201120112220_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/36/269/10636269_20210625102856_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/32/758/10632758_20210621102906_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/33/915/10633915_20210622101307_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/107/16/174/10716174_20210916153439_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/105/40/298/10540298_20201229150823_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/106/36/091/10636091_20210624104623_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm2/album/images/107/03/942/10703942_20210917110116_500.jpg/melon/resize/120/quality/80/optimize', 'https://cdnimg.melon.co.kr/cm/album/images/012/86/252/1286252_500.jpg/melon/resize/120/quality/80/optimize']

STEP2. Download images using URLs

자신의 디렉토리에 img 폴더 생성하고 실행

In [136]:
```
import urllib.request

count = 0
for link in link_list:#limk_list의 이미지 모으기
    count += 1
    urllib.request.urlretrieve(link, 'C:/Users/gksquf/OneDrive/Documents/CDesing/img/
    #img파일을 생성하고 생성한 파일에 이미지 저장
```