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 [101]:
```
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 [102]:
```
# 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 [103]:
```
# 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 확인 API" 접근해서 정보를 추출하는 프로그램

In [104]:
```
#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=180.67.72.131
REMOTE_HOST=180.67.72.131
REMOTE_PORT=34302
HTTP_HOST=api.aoikujira.com
HTTP_USER_AGENT=Python-urllib/3.7
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 (http://pypi.python.org/pypi))
- pip install beautifulsoup4
- 예제 HTML >

```
<html><body>
<h1>스크레이핑이란?</h1>
<p>웹 페이지를 분석하는 것</p>
<p>원하는 부분을 추출하는 것</p>
</body></html>
```

패키지 import 및 예제 HTML

In [105]:
```
from bs4 import BeautifulSoup
```

In [106]:
```
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 [107]:
```
soup = BeautifulSoup(html, 'html.parser')
```

3) 원하는 부분 추출하기

In [108]:
```
h1 = soup.html.body.h1
p1 = soup.html.body.p
p2 = p1.next_sibling.next_sibling
```

4) 요소의 글자 출력하기

In [109]:
```
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 [110]:
```
soup = BeautifulSoup(html, 'html.parser')
```

- 1) find() 메서드로 원하는 부분 추출하기

In [111]:
```
title = soup.find("h1")
body  = soup.find("p")
print(title)
```

```
<h1>스크레이핑이란?</h1>
```

- 2) 텍스트 부분 출력하기

In [112]:
```
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 [113]:
```
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 [114]:
```
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 [115]:
```
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 (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.

In [116]:
```
import pandas as pd
col = ['서식','설명']
con = [['*','모든 요소를 선택'], ['<요소 이름>','요소 이름을 기반으로 선택'], [
'<클래스 이름>','클래스 이름을 기반으로 선택'],['<id 이름>','id 속성을 기반으로 선
택']]
pd1 = pd.DataFrame(con,columns=col)
pd1
```

Out[116]:

	서식	설명
0	*	모든 요소를 선택
1	<요소 이름>	요소 이름을 기반으로 선택
2	<클래스 이름>	클래스 이름을 기반으로 선택
3	<id 이름>	id 속성을 기반으로 선택

BeautifulSoup에서 Css Selector 사용하기

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

In [117]:
```
col = ['메서드','설명']
ind = ['*','*']
con = [['soup.select_one(선택자)','CSS 선택자로 요소 하나를 추출합니다.'],[
'soup.select(선택자)','CSS 선택자로 요소 여러 개를 리스트를 추출합니다.']]
pd3 = pd.DataFrame(con,columns=col,index=ind)
pd3.set_index('메서드', inplace=True)
pd3
```

Out[117]:

메서드	설명
soup.select_one(선택자)	CSS 선택자로 요소 하나를 추출합니다.
soup.select(선택자)	CSS 선택자로 요소 여러 개를 리스트를 추출합니다.

In [118]:
```
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 [119]:
```
# 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)

1. html 분석 및 원하는 데이터를 추출 (BeautifulSoup)

In [120]:
```
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/ (https://finance.naver.com/marketindex/)
- 다음은 원/달러 환율 정보를 추출하는 프로그램임 ## 1) HTML 가져오기

In [121]:
```
url = "https://finance.naver.com/marketindex/"
res = request.urlopen(url)
```

2) HTML 분석하기

In [122]:
```
soup = BeautifulSoup(res, "html.parser")
```

3) 원하는 데이터 추출하기

In [123]:
```
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 (http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp)
- 참고: 기상청 RSS http://www.kma.go.kr/weather/lifenindustry/service_rss.jsp (http://www.kma.go.kr/weather/lifenindustry/service_rss.jsp)

In [124]:
```
col = ['매개변수','의미']
con = [['stnid','기상정보를 알고 싶은 지역을 지정']]
pd7 = pd.DataFrame(con,columns=col)
pd7.set_index('매개변수', inplace=True)
pd7
```

Out[124]:

매개변수	의미
stnid	기상정보를 알고 싶은 지역을 지정

- 지역번호는 다음과 같음

In [125]:
```
col = ['지역','지역번호','지역','지역번호']
con = [['전국','108','전라북도','146'],['서울/경기도','109','전라남도','15
6']
      ,['강원도','105','경상북도','143'],['충청북도','131','경상남도','159'
],
      ['충청남도','133','제주특별자치도','184']]
pd9 = pd.DataFrame(con,columns=col)
pd9
```

Out[125]:

	지역	지역번호	지역	지역번호
0	전국	108	전라북도	146
1	서울/경기도	109	전라남도	156
2	강원도	105	경상북도	143
3	충청북도	131	경상남도	159
4	충청남도	133	제주특별자치도	184

- 파이썬으로 요청 전용 매개변수를 만들 때는 urllib.parse 모듈의 urlencode() 함수를 사용해 매개변수를 URL로 인코딩한다.

1) HTML 가져오기

In [126]:
```
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=1
09
```

2) HTML 분석하기

In [127]:
```
soup = BeautifulSoup(res, "html.parser")
```

3) 원하는 데이터 추출하기

In [128]:
```
header = soup.find("header")

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

print(title)
print(wf)
```

```
서울,경기도 육상중기예보
○ (강수) 10월 6일(수)은 비가 내리겠습니다.<br />○ (기온) 이번 예보기간 아침최저기온은 13~19도, 낮최고기온은 23~28도로 오늘(26일, 아침최저기온 16~20도, 낮최고기온 26~27도)과 비슷하거나 조금 낮겠습니다.<br />○ (해상) 서해중부해상의 물결은 0.5~2.0m로 일겠습니다.
```

- css selector 기반

In [129]:
```
title = soup.select_one("header > title").text
wf = header.select_one("header wf").text

print(title)
print(wf)
```

```
서울,경기도 육상중기예보
○ (강수) 10월 6일(수)은 비가 내리겠습니다.<br />○ (기온) 이번 예보기간 아침최저기온은 13~19도, 낮최고기온은 23~28도로 오늘(26일, 아침최저기온 16~20도, 낮최고기온 26~27도)과 비슷하거나 조금 낮겠습니다.<br />○ (해상) 서해중부해상의 물결은 0.5~2.0m로 일겠습니다.
```

4.3. 윤동주 작가의 작품 목록

- 위키문헌 (https://ko.wikisource.org/wiki (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%BC (https://ko.wikisource.org/wiki/%EC%A0%80%EC%9E%90:%EC%9C%A4%EB%8F%99%EC%A3%BC))
- 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 [130]:

```
# 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%E
B%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 [131]:
```
from bs4 import BeautifulSoup
from urllib import request
```

1. 네이버 뉴스 헤드라인

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

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

In [132]:
```
url = "https://news.naver.com/"

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):
    title = a.text
    print(title)
```

```
이재명, 전북 경선서 54.55%로 승리…이낙연은 38.48%

美, 화웨이 부회장 석방…미중 갈등 해소 돌파구 되나

與 "남북 대화 파란불…北, 공동연락사무소 재설치 대화 나서야"

사세행, 공수처에 곽상도 뇌물 고발 예정

신용카드 캐시백 내수 활성화 `무용론`
솔솔
```

2. 시민의 소리 게시판

다음은 서울시 대공원의 시민의 소리 게시판 입니다.

https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgList.do?pgno=1
(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 [133]:
```
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)
```

```
['강창수님 ?오!!', '그늘막 텐트 문의', '감사 인사 드립니다.', '어린이대공원 매점 냉장고 점검 부탁드립니다.', '어린이를 위한 공원내 식당에 아기를 위한 시설 부족(아기의 자가 왜 없죠?)', '강창수 해설사님 ', '동물해설사님 칭찬', '강창수 동물 해설사님', '놀이동산 푸드코트 김치가 중국산인 이유는?', '주슨트 설명 최고예요!!'] ['https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210926000004&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210926000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210926000001&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210925000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210923000005&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210920000001&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210919000004&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210919000003&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210918000002&pgno=1', 'https://www.sisul.or.kr/open_content/childrenpark/qna/qnaMsgDetail.do;jsessionid=VDwjk481CigWLut2JrVXdOLmOOWuSjlnsSgQ9IfaTgaXxpMDplWDoBDpACwhzSab.etisw2_servlet_user?qnaid=QNAS20210909000001&pgno=1']
```

추가 내용

수집된 자료를 데이터프레임으로 만들어 csv로 저장하는 것이 일반적입니다.

In [134]:
```
import pandas as pd


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

Out[134]:

	title	link
0	강창수님 ? 오!!	https://www.sisul.or.kr/open_content/childrenp...
1	그늘막 텐트 문의	https://www.sisul.or.kr/open_content/childrenp...
2	감사 인사 드립니다.	https://www.sisul.or.kr/open_content/childrenp...
3	어린이대공원 매점 냉장고 점검 부탁드립니다.	https://www.sisul.or.kr/open_content/childrenp...
4	어린이를 위한 공원내 식당에 아기를 위한 시설 부족(아기의자가 왜 없죠?)	https://www.sisul.or.kr/open_content/childrenp...

In [186]:
```
board_df.to_csv("board.csv", index=False, encoding='utf-8-sig')
```

(Optional) 웹 크롤링2 - Dynamic Crawling

0. 라이브러리

In [136]:
```
! pip install selenium
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
```

```
Requirement already satisfied: selenium in c:\users\user\anaconda3\l
ib\site-packages (3.141.0)
Requirement already satisfied: urllib3 in c:\users\user\anaconda3\li
b\site-packages (from selenium) (1.24.2)
```

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 (https://www.melon.com/chart/index.htm)

In [168]:
```
driver = webdriver.Chrome('./chromedriver.exe')
```

In [172]:
```
 # chrome driver 설정
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).until(EC.presence_of_element_locat
ed((By.CSS_SELECTOR,'#frm > div> table > tbody > tr:nth-child(1),tr:
nth-child(4) > div >div')))

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

title.text
```

Out[172]:
```
'1\nSTAY\nThe Kid LAROI, Justin Bieber\nStay\n좋아요\n152,615'
```

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 [170]:
```
# 2번째 제목 크롤링
WebDriverWait(driver, 20) \
    .until(EC.presence_of_element_located((By.XPATH, '//*[@id="lst5
0"]/td[6]/div/div'))).text
```

Out[170]:
```
'STAY\nThe Kid LAROI, Justin Bieber'
```

실습 아주대 이비즈니스학과 강주영 교수님 text추출

In [183]:
```
driver.implicitly_wait(10)

url = "https://biz.ajou.ac.kr/ebiz/professor/professor01.jsp"

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

title3 = WebDriverWait(driver, 20).until(EC.presence_of_element_loca
ted((By.CSS_SELECTOR,'#jwxe_main_content > div > div.img_wrap > div
 > div:nth-child(2) > div:nth-child(1) > dl > dd.title > a')))

print(title3.text+ ' 교수님 a+주세요')
```

강주영 교수님 a+주세요

실습 강주영 교수님 경력 추출

In [185]:
```
driver.implicitly_wait(10)

url = "https://biz.ajou.ac.kr/ebiz/professor/professor01.jsp?include
=view&article_no=200511197&board_wrapper=%2Febiz%2Fprofessor%2Fprofe
ssor01.jsp&pager.offset=0&board_no=331"

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

title100 = WebDriverWait(driver, 20).until(EC.presence_of_element_lo
cated((By.CSS_SELECTOR,'#jwxe_main_content > div > div.view_wrap > t
able > tbody > tr:nth-child(4) > td')))

title100.text
```

Out[185]:
```
'현 아주대학교 정교수\n현 경영대학 학장, 글로벌 경영학과 학과장, 경영대학원 e러닝센터
장\n현 경영빅데이터센터 (http://bigdata.ajou.ac.kr) 센터장, 경영연구소 소장\n전
국제대학원 국제경영학과 학과장, e- 비즈니스학과장\n전 Visiting Professor of Des
autels Faculty of Management, McGill University\n현 (주) SBS Fellow
교수\n현 경기도청 빅데이터 위원회 위원\n현 한국문화정보원 빅데이터 플랫폼 미래혁신 포럼
위원\n현 도로교통공단 스마트미래교통 자문단\n현 행정안전부 공공데이터 제공 운영실태 평
가단 위원\n현 한국지능정보시스템학회  부회장\n현 한국경영정보학회 부회장\n현 한국빅데이
터학회지 편집위원장\n현 한국경영학화 경영학연구 융합분야 AE 및 편집위원 \n현 한국경영
학회 이사\n현 한국스마트미디어학회 상임이사\n현 전자거래학회 학술이사\n현 한국빅데이터
학회 이사\n현 한국블록체인경영학회 이사\n현 한국IT서비스학회 이사\n현 한국지능정보시스
템학회 이사\n현 정보시스템 연구 편집위원\n현 한국IT서비스학회지 편집위원\n현 경기도교
육청 정보화위원회 위원\n전 한국연구재단 비상근 전문위원(PM) (정보·융합기술/정보 및 지
능 분야)\n전 한국은행 IT예산편성 자문위원\n전 한국재정정보원 재정정보활용 자문위원\n전
한국경영학회 KBR 저널 운영위원장\n전 한국콘텐츠학회 홍보위원\n전 한국과학기술정책관리연
구소 위촉연구원\n전 아시안사인 CTO\n국내 정보시스템 관련 학술대회 위원장, 좌장 및 심
사위원 등 다수 역임\n공공기관 및 유관기관 자문 및 평가위원 다수 역임.'
```

강의 내용 정리

잠재의미분석의 활용

- 문서 간의 유사도
- Count vector나 TFIDF에 cosine similarity를 직접 적용하는 경우, 물리적인 단어로만 유사도를 측정하게 됨
- 잠재의미분석을 활용하면 직접적인 단어가 아니라 의미 적으로 유사한(문서에서 함께 많이 등장한) 단어들로 유사 도를 측정하는 것이 가능할 것으로 기대
- 단어 간의 유사도
- 마찬가지로 주어진 문서 집합에서 단어들이 어떤 유사도 를 가지는 지 볼 수 있음

Zipf's law(멱법칙)

- 극히 소수의 데이터가 결 정적인 영향을 미치게 됨 ### 해결방안
- feature selection
- 빈도 높은 단어를 삭제
- 심한 경우 50% 삭제
- Boolean BOW 사용
- 1이상이면 1로 변환
- log 등의 함수를 이용해 weight를 변경

텍스트 마이닝 방법

- NLP(Natural Language Processing) 기본도구
- Tokenize, stemming, lemmatize
- Chunking
- BOW, TFIDF – sparse representation ### 머신러닝(딥러닝)
- Naïve Bayes, Logistic egression, Decision tree, SVM
- Embedding(Word2Vec, Doc2Vec) – dense representation
- RNN(LSTM), Attention, Transformer

In [ ]: