In [2]:

```
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 as bts
import re #정규식 표현을 위한 모듈
import warnings
import pandas as pd
warnings.filterwarnings('ignore')

#윈도우용 크롬 웹 드라이버 실행 경로(window)지정
executable_path="chromedriver.exe"
driver=webdriver.Chrome(executable_path=executable_path)

#사이트의 html구조에 기반하여 크롤링을 수행
source_url="https://ko.wikipedia.org/wiki/%ED%8A%B9%EC%88%98:%EC%B5%9C%EA%B7%BC%EB%B0%94%EB%80%9C?hi
driver.get(source_url)

#element=WebDriverWait(driver,5).until(EC.presence_of_element_located((By.CLASS_NAME,"app"))) => 요청
req=driver.page_source

soup=bts(req,'html.parser')

atags=soup.select('.mw-title a')#제목과 url주소가 모두 들어있는 a태그 추출

base_url='https://ko.wikipedia.org'#앞에 기본으로 붙는 wikipedia 주소

page_urls=[]
for a in atags:
    print(a.text)
    page_urls.append(base_url+a['href'])#기본주소 + 추출한 주소 하여 urls에 저장
    print(base_url+a['href'])
    print('============================')
```

```
https://ko.wikipedia.org/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:%EC%9C%84%ED%8
2%A4%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8/%EC%A0%9C%EC%95%88 (https://ko.wikipedia.
org/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:%EC%9C%84%ED%82%A4%ED%94%84%EB%A1%9
C%EC%A0%9D%ED%8A%B8/%EC%A0%9C%EC%95%88)
============================
한국방송 성우극회
https://ko.wikipedia.org/wiki/%ED%95%9C%EA%B5%AD%EB%B0%A9%EC%86%A1_%EC%84%B1%EC%9
A%B0%EA%B7%B9%ED%9A%8C (https://ko.wikipedia.org/wiki/%ED%95%9C%EA%B5%AD%EB%B0%A9%
EC%86%A1_%EC%84%B1%EC%9A%B0%EA%B7%B9%ED%9A%8C)
============================
아침마당의 에피소드 목록 (2022년)
https://ko.wikipedia.org/wiki/%EC%95%84%EC%B9%A8%EB%A7%88%EB%8B%B9%EC%9D%98_%EC%9
7%90%ED%94%BC%EC%86%8C%EB%93%9C_%EB%AA%A9%EB%A1%9D_(2022%EB%85%84) (https://ko.wik
ipedia.org/wiki/%EC%95%84%EC%B9%A8%EB%A7%88%EB%8B%B9%EC%9D%98_%EC%97%90%ED%94%BC%E
C%86%8C%EB%93%9C_%EB%AA%A9%EB%A1%9D_(2022%EB%85%84))
============================
사용자토론:밀크맛 우유
https://ko.wikipedia.org/wiki/%EC%82%AC%EC%9A%A9%EC%9E%90%ED%86%A0%EB%A1%A0:%EB%B
0%80%ED%81%AC%EB%A7%9B_%EC%9A%B0%EC%9C%A0 (https://ko.wikipedia.org/wiki/%EC%82%A
C%EC%9A%A9%EC%9E%90%ED%86%A0%EB%A1%A0:%EB%B0%80%ED%81%AC%EB%A7%9B_%EC%9A%B0%EC%9C%
```

In [35]:

```
columns = ["title", "category", "content_text"]#df의 column명이될 리스트
df = pd.DataFrame(columns=columns)#df생성

for i in range(10):
    excutable_path = "chromedriver.exe"
    driver = webdriver.Chrome(executable_path=excutable_path)
    driver.get(page_urls[i]) #urls에 저장된 i번째 주소로 요청보냄
    req = driver.page_source
    soup = bts(req, 'html.parser')#beautifulsoup으로 파싱
    contents_table = soup.find(name="main")  #soup에서 main 추출

    ### 타이틀 추출
    title = contents_table.find_all('h1')[0] #main에서 h1태그 추출
    if title is not None:
        row_title = title.text.replace("₩n", " ")
    else:
        row_title = ""

    ### 카테고리 추출
    # 카테고리 정보가 없는 경우를 확인합니다.
    if len(contents_table.select("div#mw-normal-catlinks")) > 0: #카테고리 정보가 있는 div 추출
        category=contents_table.select("div#mw-normal-catlinks")[0]
    else:
        category = None

    if category is not None:
        row_category = category.text.replace(" ", "/")#카테고리별로 구분해주기 위해 공백을 /로 repla
    else:
        row_category = ""

    ### 내용 추출
    #contents_table.find_all(name="div", attrs={"class":"wiki-paragraph"})
    #div  태그 중  class 속성값이  wiki-paragraph인 요소를 추출
    content_paragraphs = contents_table.select("div.mw-parser-output > p")#내용 단락 div 추출
    #  내용으로 추출한 리스트를 하나의 문자열로 전처리
    content_corpus_list = [] # 내용 중 텍스트만 담을 빈 리스트 생성

    # content_paragraphs 리스트의 값을 순서대로 paragraphs에 대입
    if content_paragraphs is not None:
        for paragraphs in content_paragraphs:
            if paragraphs is not None:
                content_corpus_list.append(paragraphs.text.replace("₩n", " "))
            else:
                content_corpus_list.append("")
    else:
        content_corpus_list.append("")

    # 모든 정보를 하나의 데이터 프레임에 저장하기 위해서 시리즈 생성
    # 각 페이지의 정보를 추출하여 제목, 카테고리, 내용 순으로 행을 생성
    row = [row_title, row_category, "".join(content_corpus_list)]
    # 시리즈로 만듦
    series = pd.Series(row, index=df.columns)
    # 데이터 프레임에 시리즈를 추가, 한 페이지 당 하나의 행 추가
    df = df.append(series, ignore_index=True)

    # 크롤링에 사용한 브라우저를 종료합니다.
    driver.close()
```

In [36]:

```
df
```

Out[36]:

	title	category	content_text
0	천궈쿤	분류:/1975년/출생살아있는/사람홍콩의/남자/텔레비전/배우홍콩의/남자/영화/배우	천궈쿤(陳國坤, 1975년 8월 1일 ~)은 홍콩의 배우이다. 원래 안무가였는데 ...
1	라우터브루넨	분류:/스위스의/도시베른주베르너/오버란트	라우터브루넨(독일어: Lauterbrunnen)는 스위스 베른주에 위치한 도시로, ...
2	임설	분류:/1964년/출생살아있는/사람홍콩의/남자/영화/배우홍콩의/남자/가수홍콩의/남자...	임설(Lam Suet, 1964년 7월 8일 ~)은 중화인민공화국의 가수, 배우,...
3	계산 불가능 서수	분류:/순서수	집합론에서 계산 불가능 서수는 모든 계산 가능한 서수들의 상한인 서수입니다. 그러므...
4	KBS 1TV 일일 드라마	분류:/한국방송공사의/텔레비전/드라마한국방송공사/1TV/일일연속극	KBS 1TV 일일 드라마는 KBS 1TV에서 매주 평일 밤 8시 30분에 방송 중...
5	계산 불가능 서수	분류:/순서수	집합론에서 계산 불가능 서수는 모든 계산 가능한 서수들의 상한인 서수입니다. 그러므...
6	위키백과:위키프로젝트/제안	분류:/위키프로젝트	답변 아니요 위키프로젝트에요 __ Xoghks (사문) 2021년 8월 24...
7	한국방송 성우극회	분류:/대한민국의/성우/단체한국방송공사의/성우한국방송공사	한국방송 성우극회(韓國放送 聲優劇會)는 한국방송공사에서 운영하는 대한민국 방송의 성...
8	아침마당의 에피소드 목록(2022년)	분류:/아침마당의/에피소드/목록2022년/텔레비전/에피소드2022년/대한민국	이 문서에서는 대한민국의 한국방송공사 KBS 1TV의 아침 정보 프로그램《아침마당...
9	USER TALK:밀크맛 우유		메뤼 쿠뤼스뫄스 메리 크리스마스 게임 에디터돈 참가하고 싶습니다. 그런데 만약 ...