Web Scrapping

Scrapping

install packages(Library)

In [ ]:
```
pip install beautifulsoup4
pip install selenium
pip install tqdm
pip install pandas
```

import packages

In [ ]:
```
from bs4 import BeautifulSoup as bs
from selenium import webdriver
import time
from tqdm.notebook import trange
import pandas as pd
```

설명을 위한 세부 항목 소개

In [ ]:
```
#분석하고자 하는 데이터 열(column)을 지정
#리뷰 작성 날짜
Date = []
#리뷰 내용
Content = []
#리뷰 유익성
Helpful = []
#추천 여부
Recommend = []
```

In [ ]:
```
broswer = webdriver.Chrome("./chromedriver 2")
broswer.maximize_window()
#-----------------------데이터를 수집하고자 하는 URL-----------------------
url = "https://steamcommunity.com/app/1811260/reviews/?browsefilter=toprated&snr=1_
broswer.get(url)
```

In [ ]:
```
html = broswer.page_source
soup = bs(html, 'lxml')
print(soup)
```

In [ ]:
```
page ="page"+str(1)
print(soup.find("div", id=page))
```

In [ ]:
```
Content = []
contents = soup.find("div", id=page).find_all("div", {"class" : "apphub_CardTextCo
print(contents)

for i in contents:
    temp = str(i).find("</div>")
    p = str(i)[temp+6:-6]
    token = ['\t', '\n', '<br/>', '<b>', '</b>']
    for removeStr in token:
        p = p.replace(removeStr, "")
    Content.append(p)
```

```
print(Content)
```

In [ ]:
```
dates = soup.find("div", {'id' : page}).find_all("div", {"class" : "date_posted"})

for i in dates:
    d = str(i).replace('<div class="date_posted">Posted: ', '').replace('</div>',
    print(d)
    Date.append(d)
```

In [ ]:
```
helpfuls = soup.find("div", {'id': page}).find_all("div", class_="found_helpful")

for i in helpfuls:
    h = str(i).replace('<div class="found_helpful">', '').lstrip()
    len = h.find(' ')
    Helpful.append(h[:len])

print(Helpful)
```

In [ ]:
```
recommends = soup.find("div", {'id': page}).find_all("div", class_="title")

for i in recommends:
    r = str(i).replace('<div class="title">', '').replace('</div>', '')
    Recommend.append(r)
print(Recommend)
```

Scrapper 함수

In [ ]:
```
#분석하고자 하는 데이터 열(column)을 지정
#리뷰 작성 날짜
Date = []
#리뷰 내용
Content = []
#리뷰 유익성
Helpful = []
#추천 여부
Recommend = []
```

In [ ]:
```
def Scrapping(p_num):
    html = broswer.page_source
    soup = bs(html, 'lxml')

    page = "page"+str(p_num)

    contents = soup.find("div", id=page).find_all("div", {"class" : "apphub_CardTe

    for i in contents:
        temp = str(i).find("</div>")
        p = str(i)[temp+6:-6]
        #-----------------없애고 싶은 단어----------
        token = ['\t', '\n', '<br/>', '<b>', '</b>']
        for removeStr in token:
            p = p.replace(removeStr, "")
        Content.append(p)

    dates = soup.find("div", {'id' : page}).find_all("div", {"class" : "date_poste

    for i in dates:
        d = str(i).replace('<div class="date_posted">Posted: ', '').replace('</div
        Date.append(d)

    helpfuls = soup.find("div", {'id': page}).find_all("div", class_="found_helpfu
```

```
    for i in helpfuls:
        h = str(i).replace('<div class="found_helpful">', '').lstrip()
        len = h.find(' ')
        Helpful.append(h[:len])

    recommends = soup.find("div", {'id': page}).find_all("div", class_="title")

    for i in recommends:
        r = str(i).replace('<div class="title">', '').replace('</div>', '')
        Recommend.append(r)
```

In [ ]:
```
broswer = webdriver.Chrome("./chromedriver 2")
broswer.maximize_window()
#데이터를 수집하고자 하는 URL
url = "https://steamcommunity.com/app/1811260/reviews/?browsefilter=toprated&snr=1_
broswer.get(url)
```

In [ ]:
```
# 현재 문서 높이를 가져와서 저장
prev_height = broswer.execute_script("return document.body.scrollHeight")
p_num = 0
#-----------스크롤 횟수-------------
final_pnum = 10
#------로딩을 기다리는 시간(초)----
sec = 5

# 반복 수행
for _ in trange(final_pnum):
# 스크롤 가장 아래로 내림
    broswer.execute_script("window.scrollTo(0, document.body.scrollHeight)")
    p_num = p_num + 1

# 페이지 로딩 대기
    time.sleep(sec)
    Scrapping(p_num)

# 현재 문서 높이를 가져와서 저장
    curr_height = broswer.execute_script("return document.body.scrollHeight")
    if curr_height == prev_height:
        break

    prev_height = curr_height
```