

(1)

```
#Assignment 1
!pip install selenium
!apt install chromium-chromedriver

import time
from selenium import webdriver
import requests
from bs4 import BeautifulSoup

number = input()

url = 'https://m.dhlottery.co.kr/gameResult.do?method=allWin'
header_info = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36
Edge/18.19582'}
page = requests.get(url, headers = header_info)
page = page.text

specific_options = webdriver.ChromeOptions()
specific_options.add_argument('--headless')
specific_options.add_argument('--no-sandbox')
driver = webdriver.Chrome('chromedriver', options = specific_options)

driver.get(url)
time.sleep(1)

start_box = driver.find_element_by_name('drwNoStart') # name 으로 start_box
찾기
end_box = driver.find_element_by_name('drwNoEnd') # name 으로 end_box 찾기
start_box.send_keys(str(number))
end_box.send_keys(str(number))

search_xpath = '//*[@id="searchBtn"]/span'
search = driver.find_element_by_xpath(search_xpath)
search.click()
time.sleep(1)

soup = BeautifulSoup(driver.page_source, 'html.parser')
people = soup.find_all('strong')
# soup 을 반드시 xpath 를 이용해 search 를 한 후 만들고, find_all 해주어야 한다.
# 이유 : url 이 안 바뀌더라도 search 에 따라 html 이 바뀔 수 있는데, 이 변화를
반영해서 soup 해야하기 때문.

stronglist=[]
```

```

for element in people:
    txel = element.text
    txel = txel.strip()
    stronglist.append(txel)
pplnum = stronglist[1].split()
print(pplnum[1])
# 사람 수는 strong tag 중 두번째에 존재함.
# 따라서, 텍스트를 잘라서 만든 리스트의 [1]요소를 취해주어 사람 수를 알 수 있다.

# Assignment 2
!pip install selenium
!apt install chromium-chromedriver

import time
from selenium import webdriver
import requests
from bs4 import BeautifulSoup

year=input()
month=input()
day=input()
if int(month)<10:
    month = '0'+month
if int(day)<10:
    day= '0'+day
# 해당 사이트에는 9 월 이하, 9 일 이하는(예: 2022/5/5) 5 월 5 일이 아니라 05 월 05 일로 입력해야함.
# 하지만 입력 예시는 2022 5 5 의 같은 입력은 5 를 05 의 string 으로 바꿔주는 작업이 필요.

url =
'https://www.nielsenkorea.co.kr/tv_terrestrial_day.asp?menu=Tit_1&sub_menu=1_1&area=01'
header_info = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 Edge/18.19582'}
page = requests.get(url, headers = header_info)
page = page.text

specific_options = webdriver.ChromeOptions()
specific_options.add_argument('--headless')
specific_options.add_argument('--no-sandbox')
driver = webdriver.Chrome('chromedriver', options = specific_options)

driver.get(url)
time.sleep(1)

year_box = driver.find_element_by_name('sYear')

```

```

month_box = driver.find_element_by_name('sMonth')
day_box = driver.find_element_by_name('sDay')
year_box.send_keys(year)
month_box.send_keys(month)
day_box.send_keys(day)
search_xpath =
'//*[@id="sub_body"]/table[1]/tbody/tr/td/form/table/tbody/tr/td[2]/img'
search = driver.find_element_by_xpath(search_xpath)
search.click()
time.sleep(1)

soup = BeautifulSoup(driver.page_source, 'html.parser')
programs = soup.find_all(class_='tb_txt')
# 'tb_txt' class 를 find_all 해주면 방영 목록이 크롤링됨.

programlist = []
for program in programs:
    protxt = program.text
    protxt = protxt.strip()
    programlist.append(protxt)
# 이 방영 목록을 txt 로 자르고 strip 하여 programlist 에 append 시킨 후
print(programlist[0])
print(programlist[1])
print(programlist[2])
# 처음 세 개를 print 해주면 됨.

# Assignment 3
import pandas as pd

df = pd.read_csv('score.csv')
highstd= int(df.shape[0]*0.75)-1
lowstd= int(df.shape[0]*0.25)+1
# 상위 25 프로, 하위 25 프로에 해당되는 학생의 점수는 quantile 과는 다르다.
# 따라서, n%에 해당되는 학생의 점수는 위와 같은 방법으로 index 를 추출 후
scorelist=sorted(list(df['TotalScore']))
high=scorelist[highstd]
low=scorelist[lowstd]
isangchi = (high - low)*1.5
# scorelist 에서 추출한 index 의 score 로 점수를 파악

cond1 = df['TotalScore'] >= high+isangchi
cond2 = df['TotalScore'] <= low-isangchi
cond = cond1 | cond2
df_new = df[cond]
# 두 조건의 or 조건으로 새로운 dataframe 을 만들고 (이 때 성적은 매우 높거나 매우 낮거나)
for i in range(0, df_new.shape[0]):

```

```

    if df_new.iloc[i, 4]<=low-isangchi: # 낮은 경우에는 '할 수 있어요', 그 외에는
'대단합니다'출력
        print("{} 학생 할 수 있어요!".format(df_new.iloc[i,0]))
    else:
        print("{} 학생 대단합니다!".format(df_new.iloc[i,0]))

# Assignment 4
!pip install konlpy

from konlpy.tag import Komoran
text = input()
filter = ['~', '!', '@', '*', '%', '^', '&', '?']
for i in filter:
    text=text.replace(i, '')
# 특수문자를 제거하는 전처리 적용

komoran = Komoran()
result = komoran.morphs(text)
pos_result = komoran.pos(text)
# text를 형태소 단위로 분리, tagging. 이 때 pos_result list의 각요소는 (word,
품사)의 tuple
poslist=[]
for i in pos_result:
    word, pos = i # pos_result의 각 요소 tuple을 word, pos로 저장.
    if pos in ['NNG', 'NNP', 'NNB', 'NP', 'NR', 'MAG', 'MAJ', 'SN']: #pos가 본
리스트(명사, 대명사, 부사, 숫자) 안에 있을 경우
        poslist.append(word) #word를 새로운 리스트에 넣어줌
print(sorted(poslist)) # 오름차순으로 정리된 poslist를 출력.

# Assignment 5
import pandas as pd
import matplotlib.pyplot as plt

# (1)
df = pd.read_excel('job.xlsx')
cond1=df['자치구']=='노원구'
df_1 = df[cond1]
# 노원구에 해당되는 dataframe만 추출
df_1.plot('기간', '취업률') # 이 새로운 dataframe의 기간과 취업률을 plotting

# (2)
cond21 = df['기간']==2014
cond22 = df['자치구'] == '용산구'
cond23 = df['자치구'] == '서초구'
cond24 = df['자치구'] == '강남구'
cond2 = cond21 & (cond22 | cond23 | cond24)
# 2013년의 용산구+서초구+강남구의 condition cond2로 지정
df_2 = df[cond2]

```

```
# 해당 cond2 대로 filtering 하여 df_2 로 지정하고
df_2.plot('자치구', '취업자 수', kind='bar')
# bar 형태로 자치구 별 취업자 수를 fitting

# (3)
cond3 = df['기간']==2017
df_3 = df[cond3]
# 2017 년의 dataframe 을 df_3 에 저장하고
df_3.plot('졸업자 수', '취업자 수', kind='scatter')
# df_3 의 졸업자 수와 취업자 수를 scatter 형으로 fitting.
```

(2)는 소스코드에 주석으로 표시하였습니다.

(3)

```
25
26 search_xpath = '//*[@id="searchBtn"]/span'
27 search = driver.find_element_by_xpath(search_xpath)
28 search.click()
29 time.sleep(1)
30
31 soup = BeautifulSoup(driver.page_source, 'html.parser')
32 people = soup.find_all('strong')
33
34 stronglist=[]
35 for element in people:
36     txel = element.text
37     txel = txel.strip()
38     stronglist.append(txel)
39 pplnum = stronglist[1].split()
40 print(pplnum[1])
```

999  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:21: DeprecationWarning: find\_element\_  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:22: DeprecationWarning: find\_element\_  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:27: DeprecationWarning: find\_element\_  
19

그림 1 Question 1 Result

```
32 day_box.send_keys(day)
33 search_xpath = '//*[@id="sub_body"]/table[1]/tbody/tr/td/form/table/tbody/tr/td[2]/img'
34 search = driver.find_element_by_xpath(search_xpath)
35 search.click()
36 time.sleep(1)
37
38 soup = BeautifulSoup(driver.page_source, 'html.parser')
39 programs = soup.find_all(class_='tb_txt')
40 programlist = []
41 for program in programs:
42     protxt = program.text
43     protxt = protxt.strip()
44     programlist.append(protxt)
45 print(programlist[0])
46 print(programlist[1])
47 print(programlist[2])
```

2022  
6  
1  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:27: DeprecationWarning: find\_eleme  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:28: DeprecationWarning: find\_eleme  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:29: DeprecationWarning: find\_eleme  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:34: DeprecationWarning: find\_eleme  
KBS9시뉴스  
아침마당  
인간극장

그림 2 Question 2 Result



```
1 # Assignment 3
2
3 import pandas as pd
4
5 df = pd.read_csv('score.csv')
6 highstd= int(df.shape[0]*0.75)-1
7 lowstd= int(df.shape[0]*0.25)+1
8 scorelist=sorted(list(df['TotalScore']))
9 high=scorelist[highstd]
10 low=scorelist[lowstd]
11 isangchi = (high - low)*1.5
12
13 cond1 = df['TotalScore'] >= high+isangchi
14 cond2 = df['TotalScore'] <= low-isangchi
15 cond = cond1 | cond2
16 df_new = df[cond]
17 for i in range(0, df_new.shape[0]):
18     if df_new.iloc[i, 4]<=low-isangchi:
19         print("{} 학생 할 수 있어요!".format(df_new.iloc[i,0]))
20     else:
21         print("{} 학생 대단합니다!".format(df_new.iloc[i,0]))
```

2021826381 학생 할 수 있어요!  
2017936402 학생 대단합니다!  
2019396026 학생 대단합니다!  
2021029260 학생 할 수 있어요!

그림 3 Question 3 Result



```
1 from konlpy.tag import Komoran
2 text = input()
3 filter = ['~', '!', '@', '*', '%', '^', '&', '?']
4 for i in filter:
5     text=text.replace(i, '')
6
7 komoran = Komoran()
8 result = komoran.morphs(text)
9 pos_result = komoran.pos(text)
10 poslist=[]
11 for i in pos_result:
12     word, pos = i
13     if pos in ['NNG', 'NNP', 'NNB', 'NP', 'NR', 'MAG', 'MAJ', 'SN']:
14         poslist.append(word)
15 print(sorted(poslist))
```

한일 양국 간 90일 이내 무비자 체류 제도가 코로나 19로 효력이 중단된 상황이다.  
['19', '90', '간', '비자', '상황', '양국', '이내', '일', '제도', '중단', '체류', '코로나', '한일', '효력']

그림 4 Question 4 Result

```
font.set_text(s, 0, flags=flags)
/var/local/lib/python3.7/dist-packages/matplotlib/backends/backend_agg.py:214: RuntimeWarning: Glyph 51520 missing from current font.
font.set_text(s, 0.0, flags=flags)
/var/local/lib/python3.7/dist-packages/matplotlib/backends/backend_agg.py:189: RuntimeWarning: Glyph 51520 missing from current font.
font.set_text(s, 0, flags=flags)
```

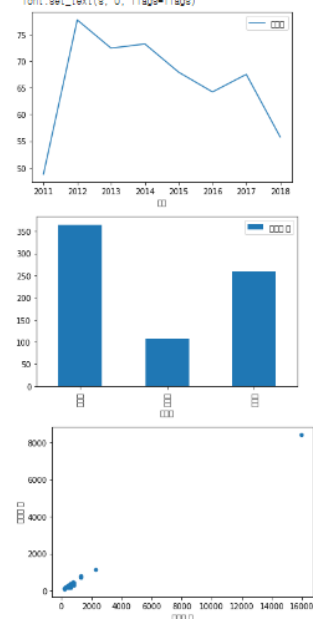


그림 5 Question 5 Result