



< 파이썬 과제 >

웹 크롤링 & 데이터 필터링

< 김소연 >





목차

- 주제 소개
- Selenium
- Pandas
- Tkinter
- 계획






과제 주제


Today Hot Light Hour

HOT NOW 등의 켜지고 따뜻하고 달콤한 도넛이 나오는 시간입니다.
현재 **7** 개의 매장에 HOT NOW 등이 켜졌습니다.


5/10(화) **9:00** 09:00 **확인**



영등포점
09:00 ~ 10:00
영업시간 월~일 08:00 ~ 22:00
[바로가기](#)



대치점
14:00 ~ 16:00
09:00 ~ 11:00
영업시간 월~일 08:00 ~ 22:00
[바로가기](#)



대전둔산점
17:00 ~ 19:00
09:00 ~ 11:00
영업시간 월~일 06:00 ~ 24:00
[바로가기](#)

▶ Selenium을 이용한 동적 웹 크롤링

크리스피크림도넛 HOTNOW 매장 시간별 알아보기

HotNow매장 : 갓 나온 따뜻한 도넛을 살 수 있는 매장

07:00 – 24:00 중 시간을 선택하고 확인 버튼을 누르면
해당 시간대에 HotNow 등이 켜진 매장을 확인할 수 있음

← 크리스피크림 hotnow 팝업창





Selenium

▶ 태그로 원하는 정보를 찾아 저장

```
def setTime(in_time):  
    #시간 설정  
    driver.find_element_by_xpath("//option[@value='" + str(in_time) + "']").click()  
    #검색btn 클릭  
    driver.find_element_by_xpath("//input[@type = 'image']").click()  
    time.sleep(2)  
  
    #매장 갯수 저장  
    store_cnt = driver.find_element_by_css_selector("span.data_txt").text  
    store_cnt = int(store_cnt)  
    print('count = ',store_cnt)  
  
    if store_cnt > 0:  
        #해당 시간에 hotnow하는 매장 정보  
        storeInfo = driver.find_elements_by_class_name("clfix")  
  
        for info in storeInfo:  
            imgURL = info.find_element_by_css_selector("span.img > img").get_attribute("src")  
            imgsave = info.find_element_by_css_selector("span.img > img").get_attribute("alt")  
            urllib.request.urlretrieve(imgURL,imgsave+'.jpg')  
            name = info.find_element_by_css_selector("div.info > span.place_name").text  
            times = info.find_element_by_css_selector("div.info > span.time").text  
            link = info.find_element_by_css_selector("div.info > a").get_attribute("href")  
            print(imgURL,imgsave,name,times,link)
```

Today Hot Light Hour

HOT NOW 등의 켜지고 따뜻하고 달콤한 도넛이 나오는 시간입니다.
현재 **7** 개의 매장에 HOT NOW 등이 켜졌습니다.

5/10(화) 9:00

09:00

확인



영등포점

09:00 ~ 10:00

영업시간 월~일 08:00 ~ 22:00

바로가기



대치점

14:00 ~ 16:00

09:00 ~ 11:00

영업시간 월~일 08:00 ~ 22:00

바로가기



대전둔산점

17:00 ~ 19:00

09:00 ~ 11:00

영업시간 월~일 06:00 ~ 24:00

바로가기





Selenium

▶ 크롤링 후 CSV파일로 저장

→ 시간, 매장 이미지, 매장명, 영업시간, 바로가기 링크

```
# csv
df = pd.DataFrame()
df['time'] = sto_cnt
df['img'] = img_li
df['imgname'] = imgname_li
df['store'] = name_li
df['times'] = time_li
df['link'] = link_li
df.to_csv('hotnow_test_img.csv',encoding='cp949')
```



	A	B	C	D	E	F	G	H	I	J
1		time	img	imgname	store	times	link			
2	0	7	NONE	NONE	NONE	NONE	NONE			
3	1	8	https://ww-		수원인계점	15:00 ~ 17:00 08:00 ~ 10:00 영업시간 월~일 09:00~ 23:00	javascript:goPage(102);			
4	2	9	https://ww	영등포점	영등포점	09:00 ~ 10:00 영업시간 월~일 08:00 ~ 22:00	javascript:goPage(6);			
5	3	9	https://ww	대치점 매?	대치점	14:00 ~ 16:00 09:00 ~ 11:00 영업시간 월~일 08:00 ~ 22:00	javascript:goPage(9);			





▶ 크롤링한 CSV파일로 해본 간단한 데이터 필터링

Q1 . HotNow 매장이 가장 많은 시간과 매장 수, 매장명은?

Q2 . HotNow 매장이 없는 시간은?

Q3 . HotNow 매장의 총개수는?





Pandas

```
#데이터 불러오기
df = pd.read_csv('hotnow_test_img.csv', encoding='cp949')

"""
Q1. hotnow 매장이 가장 많은 시간대와 매장명은?
"""
# hotnow가 많이 뜨는 시간대 저장 변수 (결과출력변수)
hotmax_time = []

#time열을 기준으로 count
data = df['time'].value_counts()
#data변수안에 값중 가장 큰 값만 출력
data_max = data.max()

#데이터 필터링 ( Q1. 7시- 24시 중에서 가장 도넛이 많이 나오는 시간은? )
filters = df.groupby('time').filter(lambda x:x['time'].count() == data_max)
filter_data = filters['time'].value_counts()

for idx in range(0,len(filter_data),1):
    #필터링된 결과값에서 인덱스만 추출
    data_idx = filter_data.index[idx]

    #결과값출력 변수에 값 저장
    hotmax_time.append(data_idx)

    # time열에서 인덱스와 같은 정보들만 추출
    reslut = df['time'] == data_idx

    #결과 출력

    print(df.loc[reslut])

print('=====')
```

Q1 . HotNow 매장이 가장 많은 시간과 매장 수, 매장 명은?





Q1 . HotNow 매장이 가장 많은 시간과 매장 수, 매장명은?

```
# hotnow가 많이 뜨는 시간대 매장 갯수와 매장명 저장 변수 (결과출력변수)
hotmax_cnt = df.loc[reslut].loc[:, 'store'].count()
hotmax_name = df.loc[reslut].loc[:, 'store'] #loc[:, 'store'] : store열 만을 대상으로 검색/추출

print('Q1. 7시 - 24시 중에서 가장 도넛이 많이 나오는 시간은? \n → {0}시 입니다. {1}개의 매장에서 도넛이 생산됩니다.'.format(hotmax_time, hotmax_cnt))
print(' → 매장명 : \n{0}'.format(hotmax_name))
```

#. 결과

```
Q1. 7시 - 24시 중에서 가장 도넛이 많이 나오는 시간은?
→ [9, 10]시 입니다. 7개의 매장에서 도넛이 생산됩니다.
→ 매장명 :
9      영등포점
10     대치점
11     대전둔산점
12     건대스타시티점
13     울산삼산점
14     수원인계점
15     원주무실점
Name: store, dtype: object
```





Pandas

```
#데이터 불러오기
df = pd.read_csv('hotnow_test_img.csv', encoding='cp949')

"""
Q1. hotnow 매장이 가장 많은 시간대와 매장명은?
"""

# hotnow가 많이 뜨는 시간대 저장 변수 (결과출력변수)
hotmax_time = []

#time열을 기준으로 count
data = df['time'].value_counts()
#data변수안에 값중 가장 큰 값만 출력
data_max = data.max()

#데이터 필터링 ( Q1. 7시- 24시 중에서 가장 도넛이 많이 나오는 시간은? )
filters = df.groupby('time').filter(lambda x:x['time'].count() == data_max)
filter_data = filters['time'].value_counts()

for idx in range(0,len(filter_data),1):
    #필터링된 결과값에서 인덱스만 추출
    data_idx = filter_data.index[idx]

    #결과값출력 변수에 값 저장
    hotmax_time.append(data_idx)

    # time열에서 인덱스와 같은 정보들만 추출
    reslut = df['time'] == data_idx

    #결과 출력

    print(df.loc[reslut])

print('=====')
```

Q2 . HotNow 매장이 없는 시간은?





Q2 . HotNow 매장이 없는 시간은?

```
"""
Q2. 7시 - 24시 중에서 hotnow 매장이 없는 시간은?
"""

#카운트
none_cnt = df.query("store == 'NONE'").loc[:, 'time'].count()
#시간대
none_time = df.query("store == 'NONE'").loc[:, 'time'].values

print('Q2. 7시 - 24시 중에서 hotnow 매장이 없는 시간은? \n → {0}시 입니다.'.format(none_time))
```

#. 결과

```
=====
Q2. 7시 - 24시 중에서 hotnow 매장이 없는 시간은?
→ [ 7 21 22 23 24]시 입니다.
=====
```





Pandas

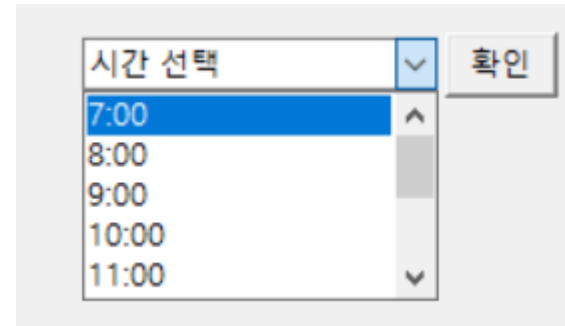
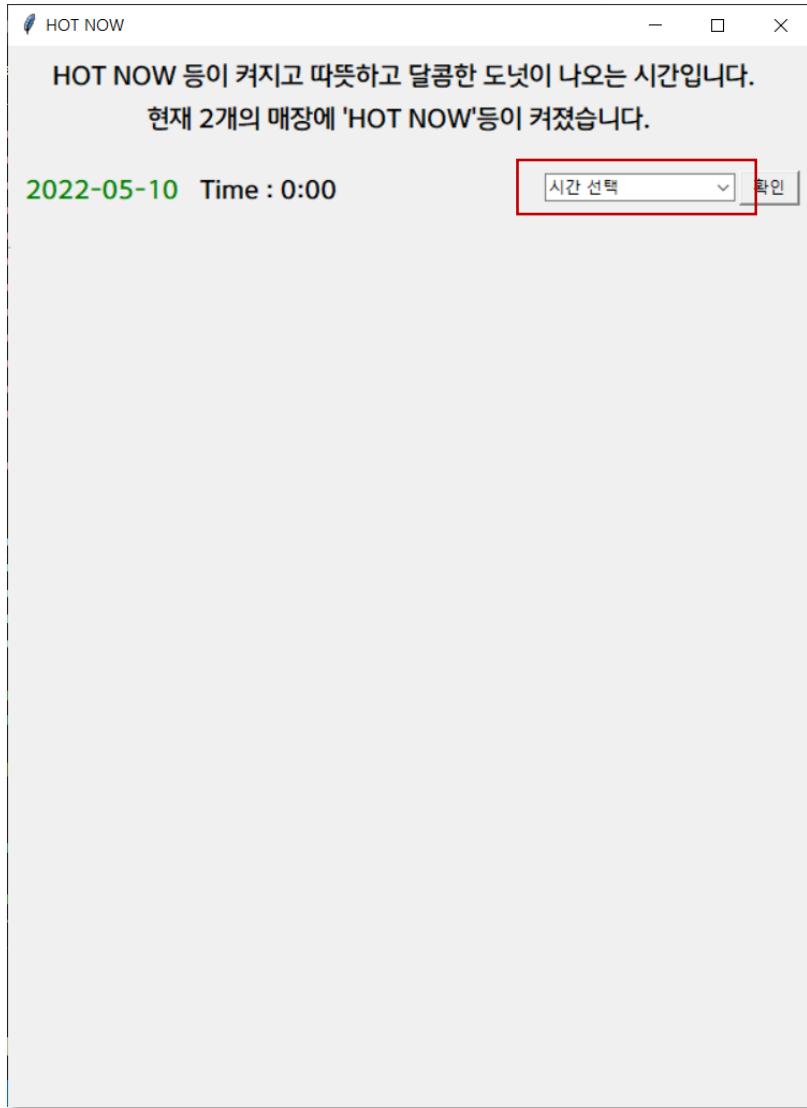
Q3 . HotNow 매장의 총개수는?

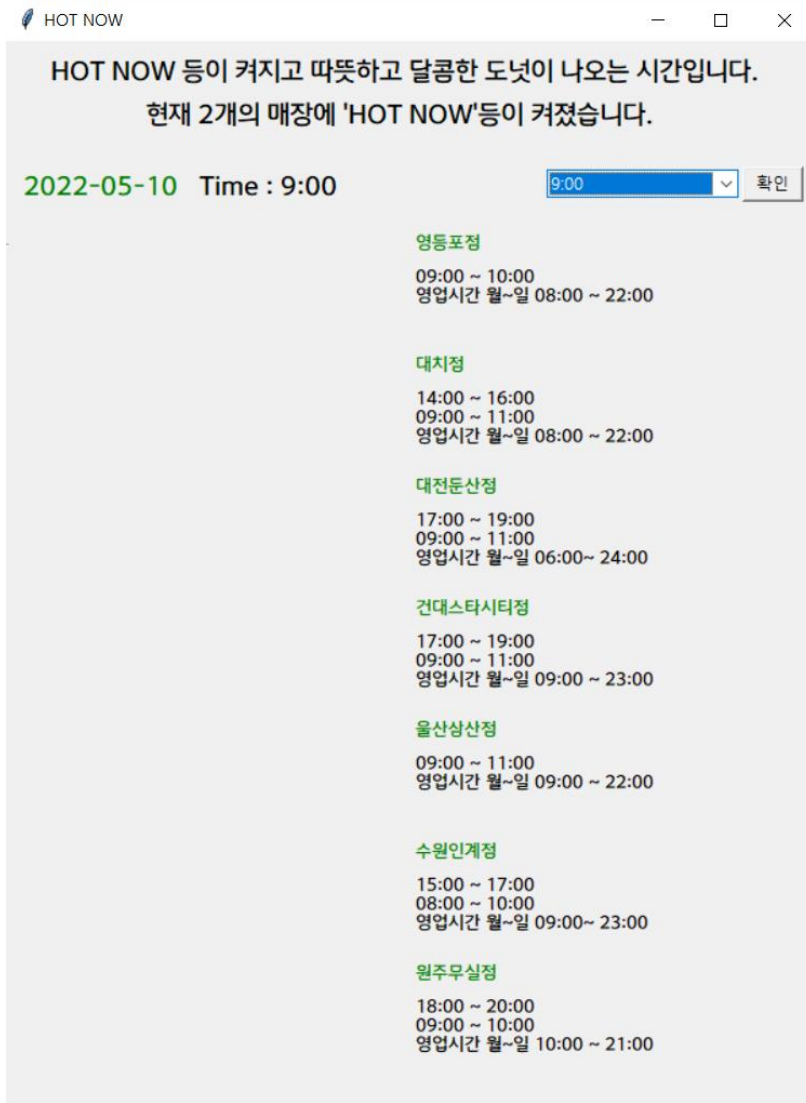
#. 결과





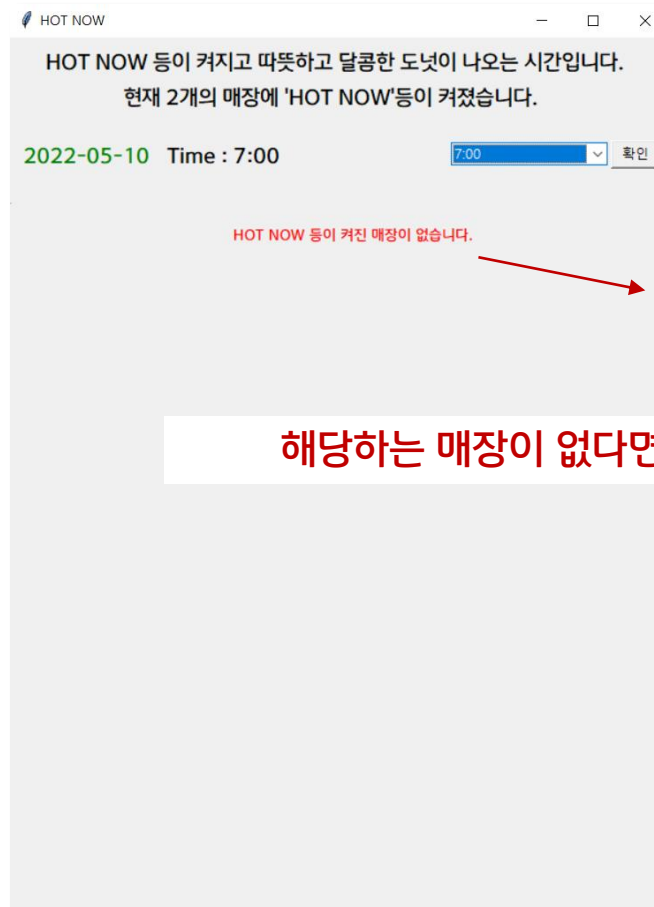
tkinter





#. 결과

내가 선택한 시간에 hotnow 매장 리스트를 확인 할 수 있음.



HOT NOW 등이 켜진 매장이 없습니다.

해당하는 매장이 없다면 위와 같이 텍스트가 출력됨.



Today Hot Light Hour

HOT NOW 등의 켜지고 따뜻하고 달콤한 도넛이 나오는 시간입니다.
현재 1 개의 매장에 HOT NOW 등이 켜졌습니다.

5/11 22:10

7:00 ▾

확인



김해봉황점

11:00-13:00

바로가기

+ html 문서와 데이터 연결하여 기능 구현

+ 이전에 구현해 놓은 html 파일과 연결

5/11 9:00

9:00 ▾

확인



인계점

11:00-13:00

바로가기



마무리

감사합니다.😊

