H.S. Kim
Christina

2020년도 홈페이지를 기반으로 크롤링한 강의 노트입니다.
홈페이지에서 달라진 코드 부분을 찾아보고,
코딩 내용을 수정하여 크롤링을 완성해봅시다!!

Web Crawling based on Image data

검색 결과를 다양한 형식으로 저장 (한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글

실습 노트 참고

- 10_Web Crawling based on Image data (한국관광공사_이미지 데이터 저장).ipynb

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

Step 1. 필요한 모듈과 함수를 로딩하고 검색어 입력받기

```
1 from bs4 import BeautifulSoup
2 from selenium import webdriver
3 import urllib
4 import time
5 import os
```

urllib는 URL 작업을 위한 여러 모듈을 모은 패키지로 4개의 모듈을 포함
1) URL을 열고 읽기 위한 urllib.request
2) urllib.request에 의해 발생하는 예외를 포함하는 urllib.error
3) URL 구문 분석을 위한 urllib.parse
4) robots.txt 파일을 구문 분석하기 위한 urllib.robotparser

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. **현재 날짜 및 시간 획득**
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

```
1  os.getcwd()
```

```
'D:₩₩ai'
```

```
1  os.chdir("D:₩₩ai₩₩DATA")
2  os.getcwd()
```

```
'D:₩₩ai₩₩DATA'
```

현재 날짜 및 시간 획득

```
1  #참고
2  now = time.localtime()
```

이름	값	비고
tm_year	연	예: 1993, 2019
tm_mon	달	범위: 1~12
tm_mday	일	범위: 1~31
tm_hour	시	범위: 0~23
tm_min	분	범위: 0~59
tm_sec	초	범위: 0~61
tm_wday	요일	범위: 0~6 (0: 월요일)
tm_yday	연중 경과일	범위: 1~366
tm_isdst	일광절약타임 적용여부	0: 미적용 1: 적용 -1: 모름

```
1  now
```

```
time.struct_time(tm_year=2020, tm_mon=10, tm_mday=9, tm_hour=16, tm_min=13, tm_sec=2, tm_wday=4, tm_yday=283, tm_isdst=0)
```

```
1  print(now.tm_year,"년", now.tm_mon,"월", now.tm_mday,"일")
2  print(now.tm_hour,"시", now.tm_min,"분", now.tm_sec,"초")
```

```
2020 년 10 월 9 일
16 시 13 분 2 초
```

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

Step 2. 파일을 저장할 디렉토리 생성하기

```
1 txt1 ='VISIT_KO_IMAGE '
2 txt2 = '%04d-%02d-%02d-%02d-%02d-%02d' %(now.tm_year, now.tm_mon, now.tm_mday, now.tm_hour, now.tm_min, now.tm_sec)
3 dir_name = txt1+txt2
4 print(dir_name)
```

```
VISIT_KO_IMAGE 2020-10-09-16-13-02
```

```
1 os.makedirs(dir_name) #디렉토리 생성
2 os.chdir(dir_name)
3 os.getcwd()
```

```
'D:₩₩ai₩₩DATA₩₩VISIT_KO_IMAGE 2020-10-09-16-13-02'
```

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

https://korean.visitkorea.or.kr/detail/rem_detail.html?cotid=be3db10c-b642-409c-81cc-c4cdecb5bd8b&temp=

Step 3. 크롬 드라이버를 사용해서 웹 브라우저를 실행하기

```
path = "c:/temp/chromedriver_240/chromedriver.exe"
driver = webdriver.Chrome(path)

s_time = time.time( )      #크롤링 시작 시간을 위한 타임 스탬프

#이미지가 포함된 웹페이지 접속
driver.get("https://korean.visitkorea.or.kr/detail/rem_detail.html?cotid=be3db10c-b642-409c-81cc-c4cdecb5bd8b&temp=")
time.sleep(2)  #페이지가 모두 열릴 때 까지 2초 대기
```

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

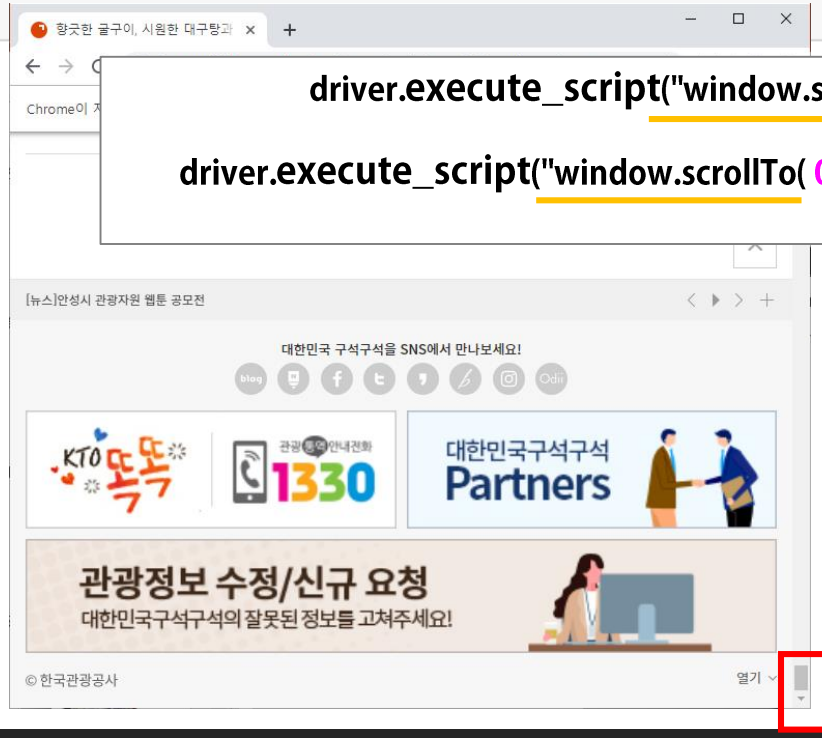
- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. **화면 자동 스크롤**
4. 이미지 수집

- 구글
- Exercise

Step 4. 자동 스크롤다운 함수를 정의한 후 호출하여 실행하기

```
1  def scroll_down(driver):
2      driver.execute_script("window.scrollTo(0,document.body.scrollHeight);") # 화면 맨 아래까지 이동
3     # driver.execute_script("window.scrollTo(0,500);")  #절대 좌표
4     # driver.execute_script("window.scrollBy(0,500);")  #상대 좌표
5      time.sleep(1)
6
7  scroll_down(driver)
```



driver.execute_script("window.scrollTo(x 좌표 , y 좌표) ;")

driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

Step 5. 웹 페이지 이미지 접근하기

```
1  count = 1
2  img_src2=[]
3
4  html = driver.page_source
5  soup = BeautifulSoup(html, 'html.parser')
6  img_src = soup.find('div','box_txtPhoto').find_all('img')  #div 태그 class명이 box_txtPhoto
```

```
1  img_src
```

```
[<img alt="굴구이" src="https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&amp;id=7ee736d9-5afa-471a-976d-42da60a69a51"/>
 <img alt="거제에서 쉽게 볼 수 있는 굴 양식장" src="https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&amp;id=2498f000-e8
da60e6b"/>,
 <img alt="굴무침" src="https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&amp;id=08ea1346-7aa5-4460-94aa-c262d05c0af8"/>
 <img alt="굴구이 먹을 때 사용하는 칼과 장갑" src="https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&amp;id=0501f778-7fe
348c2a"/>,
 <img alt="거제의 겨울 별미 굴구이" src="https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&amp;id=638ea611-96fe-465e-a3a
```

```
1  type(img_src)
```

```
bs4.element.ResultSet
```

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

```
for i in img_src :
    print("●", count,"번째")
    print("▶type(i) :", type(i))
    print("▶value(i) : ", i)

    img_src1 = i['src']  #src 태그 부분
    print("▶img_src1 :", img_src1)

    img_src2.append(img_src1)
    print("▶img_src2 :", img_src2) #src 태그 부분들이 계속해서 append됨

    count += 1
    print("-"*80)
```

```
● 1 번째
▶type(i) : <class 'bs4.element.Tag'>
▶value(i) :  <img alt="굴구이" src="https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&amp;id=7ee736d9-5afa-47
▶img_src1 : https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&id=7ee736d9-5afa-471a-976d-42da60a69a51
▶img_src2 : ['https://cdn.visitkorea.or.kr/img/call?cmd=VIEW&id=7ee736d9-5afa-471a-976d-42da60a69a51']
--------------------------------------------------------------------------------
● 2 번째
▶type(i) : <class 'bs4.element.Tag'>
▶value(i) :  <img alt="거제에서 쉽게 볼 수 있는 굴 양식장" src="https://cdn.visitkorea.or.kr/img/call?cmd=
fff-8f5c-f42feda60e6b" />
```

```
print(len(img_src2)) #img_src2는 src 태그 부분들이 계속해서 append된 리스트
```

```
14
```

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- **한국관광공사**

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. **이미지 수집**

- 구글
- Exercise

Step 6. 웹 페이지 이미지 수집하기

```
file_no = 0
for i in range(0,len(img_src2)) :
        try :
                urllib.request.urlretrieve(img_src2[i], str(file_no)+'.jpg')
                #urllib.request.urlretrieve 이미지를 다운로드하는 함수이다.
                #file_no는 파일명으로 사용하기위해서 위에서 0값을 할당해놓은 변수이다.
        except :
                print("이미지가 없습니다.")
                continue
        file_no += 1
        #file_no는 파일명으로 사용하기 위해서 위에서 할당해놓은 변수로 1씩 더해서 파일명으로 사용한다.

        time.sleep(0.5)
        print("%s 번째 이미지 저장중입니다." %file_no)
```

```
1 번째 이미지 저장중입니다.
2 번째 이미지 저장중입니다.
3 번째 이미지 저장중입니다.
4 번째 이미지 저장중입니다.
5 번째 이미지 저장중입니다.
6 번째 이미지 저장중입니다.
7 번째 이미지 저장중입니다.
8 번째 이미지 저장중입니다.
9 번째 이미지 저장중입니다.
10 번째 이미지 저장중입니다.
11 번째 이미지 저장중입니다.
12 번째 이미지 저장중입니다.
13 번째 이미지 저장중입니다.
14 번째 이미지 저장중입니다.
```

urllib.request.urlretrieve()
이미지를 다운로드 하는 함수

함수에서 사용하는 **argument**
(img_src2[i], str(file_no)+'.jpg')
① ②

①에 해당하는 이미지를 ②이름으로 저장

이미지 데이터 수집 후 저장(한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글
- Exercise

Step 7. 이미지 데이터 수집 후 요약 정보 출력하기

```
1  print(os.path.exists("D:/ai/DATA/"+dir_name) )
```

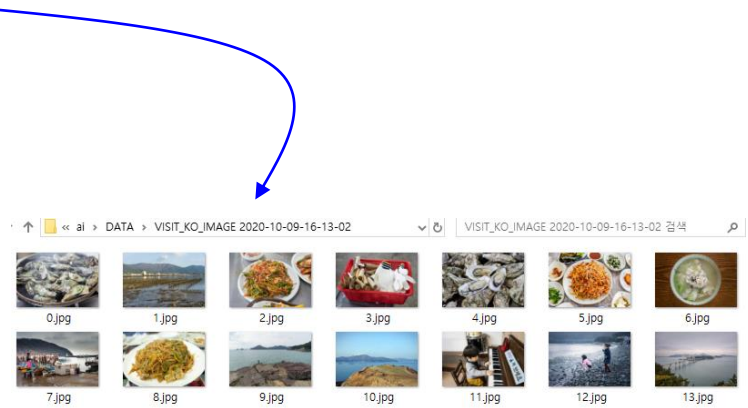
```
True
```

```
1  result_dir = "D:/ai/DATA/"+dir_name
2  print(result_dir)
```

```
D:/ai/DATA/VISIT_KO_IMAGE 2020-10-09-16-13-02
```

```
1  e_time = time.time( )
2  t_time = e_time - s_time
3
4  print("=" *70)
5  print("총 소요시간은 %s 초 입니다 " %round(t_time,1))
6  print("총 저장 건수는 %s 건 입니다 " %file_no)
7  print("파일 저장 경로: %s 입니다" %result_dir)
8  print("=" *70)
9
10 driver.close( )
```

```
======================================================================
총 소요시간은 27.5 초 입니다
총 저장 건수는 14 건 입니다
파일 저장 경로: D:/ai/DATA/VISIT_KO_IMAGE 2020-10-09-16-13-02 입니다
======================================================================
```



검색 결과를 다양한 형식으로 저장 (한국관광공사 홈페이지 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글

실습 노트 참고

- 10_Web Crawling based on Image data (구글_이미지 데이터 저장).ipynb

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

Step 1. 필요한 모듈과 함수 로딩하기

```
1 from bs4 import BeautifulSoup
2 from selenium import webdriver
3 import urllib
4 import time
5 import os
6 import math
7 import random
```

```
1 os.chdir("D:/ai/DATA")
2 os.getcwd()
```

```
'D:₩₩ai₩₩DATA'
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

Step 2. 이미지 검색 크롤러 구현에 필요한 정보 입력 받기

```
print("=" *80)
print("구글 사이트에서 이미지를 검색하여 수집하는 크롤러 입니다 ")
print("=" *80)

query_txt = input('1.크롤링할 이미지의 키워드는 무엇입니까?: ')
cnt = int(input('2.크롤링 할 건수는 몇건입니까?: (예: 120)  '))

real_cnt = math.ceil(cnt / 50) #math모듈로 부터 제공됨... 50 숫자는 임의의 숫자로 변경 가능..
#3번 정도 아래로 스크롤 다운이 이루어지도록...
#120/50 = 2.4
#math.ceil(120/50) 는 3이다.
#cnt가 120 이면 아래의 real_cnt 변수는 3 이 된다.

print("요청하신 데이터 수집을 시작하겠습니다.")
```

```
================================================================================
구글 사이트에서 이미지를 검색하여 수집하는 크롤러 입니다
================================================================================
1.크롤링할 이미지의 키워드는 무엇입니까?: 삼고양이
2.크롤링 할 건수는 몇건입니까?: (예: 120)  120
요청하신 데이터 수집을 시작하겠습니다.
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

Step 3. 이미지 수집 후 저장할 디렉토리명 셋팅하기

```
1 now = time.localtime()
2 txt1 ='GOOGLE IMAGE '
3 txt2 = '%04d-%02d-%02d-%02d-%02d-%02d' %(now.tm_year, now.tm_mon, now.tm_mday, now.tm_hour, now.tm_min, now.tm_sec)
4 dir_name = txt1+txt2
5 print(dir_name)
6
7 os.makedirs(dir_name) #디렉토리 생성
8 os.chdir(dir_name)
9 os.getcwd()
```

```
GOOGLE IMAGE 2020-10-09-15-33-58

'D:₩₩ai₩₩DATA₩₩GOOGLE IMAGE 2020-10-09-15-33-58'
```

```
1 #이미지 데이터 수집 후 저장될 디렉토리
2 result_dir = os.getcwd()
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

Step 4. 크롬 드라이버를 사용해서 웹 브라우저를 활성화시킨 후 입력창에 검색 키워드 전달하기

```
1  s_time = time.time( )
2
3  path = "c:/temp/chromedriver_240/chromedriver.exe"
4  driver = webdriver.Chrome(path)
5
6  driver.get('https://www.google.com')
7  time.sleep(random.randrange(2,5)) #random 모듈로 부터 제공되는 부분
8
9  #element = driver.find_element_by_name("q")
10 element = driver.find_element_by_xpath("""//*[@id="tsf"]/div[2]/div[1]/div[1]/div/div[2]/input""")
11
12 element.send_keys(query_txt)
13 element.submit()
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

Step 5. 이미지 메뉴에 해당하는 페이지로 진입하여 수직 스크롤바를 real_cnt 횟수만큼 스크롤다운 해보기

```
1  #driver.find_element_by_link_text("이미지").click()
2  driver.find_element_by_xpath("""//*[@id="hdtb-msb-vis"]/div[2]/a""").click()  #태그 <a 부분에 해당하는 xpath
3
4  #스크롤다운 함수 정의
5  def scroll_down(driver):
6      driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
7      time.sleep(3)
8
9  i = 1
10
11 # cnt가 120 이었다면 real_cnt는 3 이다.
12 while (i <= real_cnt): #real_cnt 보다 i값이 작거나 같을 동안 반복이므로 cnt가 120일 경우 3회 반복
13     scroll_down(driver) #위에서 정의한 scroll_down함수를 3번 호출.. 3번 아래로 스크롤 다운된다.
14     i += 1
15
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

Step 6. 이미지를 추출하여 저장하기

```
1 html = driver.page_source  #현재 페이지
2 soup = BeautifulSoup(html, 'html.parser')
3
4 imgs = driver.find_elements_by_tag_name('img') #태그명이'img'인 태그를 모두 찾는다.
5
6 #F12 키를 눌러서  ctrl+f (찾기) 기능으로 <img 부분을 확인해본다.
```

```
1 imgs
```

```
[<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-1")>,
 <selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-2")>,
 <selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-3")>,
 <selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-4")>,
```

참고
세션(session)이란 웹 사이트의 여러 페이지에 걸쳐 사용되는 사용자 정보를 저장하는 방법을 의미하며
엘리먼트(element)란 HTML 문서나 웹 페이지를 이루는 개별적인 요소를 말한다.

```
1 type(imgs)
```

```
list
```

```
1 print(len(imgs))
```

```
330
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

참고 사항

- 이미지 접근 방식

```
src는 source의 축약어이다.

<img src= 또는 data-src= 속성을 이용하여 접근할 수 있다.
src 및 data-src 속성은 서로 다르다는 점을 이해해야 한다.
data-src 속성은 HTML5 에서부터 사용되기 시작한 속성이며 대부분의 웹브라우저에서 사용할 수 있다.

<img src= 속성은 이미지가 저장되어 있는 경로를 속성값으로 요구한다.
<img data-src= 속성은 스타일 지정등에 사용하기 위해 보이지 않는 데이터를 암호화하여 사용할 수 있다.
```

```
#아래의 코드를 실행해보고 None 출력된 부분을 확인해 보자.
count = 0
img_src_test_list=[]

for img in imgs:
    print("● ", count, "번 image의 img 태그 부분 ")
    print(img)

    img_src_test=img.get_attribute('src')
    print("▶image의 src 부분 ")
    print(img_src_test)

    img_src_test_list.append(img_src_test)  #'src'태그의 value를 img_src_test_list에 추가한다.
    count += 1
    print('-'*80)

#결과를 확인해보면 20년 10월 9일의 경우 78번, 79번, 80번과 같은 곳에서 None 부분이 출력된 곳이 있다.
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

```
● 76 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-77")>
▶image의 src 부분
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcQ1mHXJwmgJN7fkamUaSYGEAb0DniBzIlQYZA&usqp=CAU
--------------------------------------------------------------------------------
● 77 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-78")>
▶image의 src 부분
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTZh-FhG3yseG1hRMyIp6H03iUwkPDp2lVlNw&usqp=CAU
--------------------------------------------------------------------------------
● 78 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-79")>
▶image의 src 부분
None
--------------------------------------------------------------------------------
● 79 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-80")>
▶image의 src 부분
None
--------------------------------------------------------------------------------
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

```
#위의 리스트 출력 결과 중에서 None 부분을 해결하기 위한 방식
#이미지 접근 방식

count = 0
final_img_src=[]

for img in imgs:
        print("● ", count, "번 image의 img 태그 부분 ")
        print(img)

        img_src1=img.get_attribute('data-src') #'img'태그의 속성'data-src' 값을 얻는다.

        if not img_src1:  # if None:
            img_src1=img.get_attribute('src')  #'img'태그의 속성'src' 값을 얻는다.

        print("▶image의 data-src 또는 src 부분")
        print(img_src1)
        final_img_src.append(img_src1)  #이미지를 final_img_src 리스트에 추가한다.

        count += 1
        print('-'*80)
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

● 76 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-77")>
▶image의 data-src 또는 src 부분
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcQ1mHXJwmgJN7fkamUaSVGEAb0DniBzIIQYZA&usqp=CAU

● 77 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-78")>
▶image의 data-src 또는 src 부분
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTZh-FhG3yseG1hRMyIp6HO3iUwkPDp2IVINw&usqp=CAU

● 78 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-79")>
▶image의 data-src 또는 src 부분
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcS6aEZPX2vttCqEVrz0y-e92N6zojt6hvAQqx7IZOWAEGBc9Y-I&usqp=CAU

● 79 번 image의 img 태그 부분
<selenium.webdriver.remote.webelement.WebElement (session="936b0f03a1d4607571b2e370902ff810", element="0.6406773499080767-80")>
▶image의 data-src 또는 src 부분
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcR6sFaeSoEu0oQ4oR4iCkSbLz08_1dnZfztI0xPUXBpvI9iD2dH&usqp=CAU

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

```
1 len(final_img_src)
```

330

final_img_src 리스트 내용을 for문을 활용하여 출력해보기

```
1 index=0
2 for i in range(0,len(final_img_src)) :
3     print("● ", index)
4     print(final_img_src[i])
5     index+=1
6
```

https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcSr_fZQeGp5twx8W3yL7VLG5kn1U4ZHqF0Sng&usqp=CAU
● 321
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRwJzT1xirwRbx3m5E24_FwEzkFPDGj-9jkXQ&usqp=CAU
● 322
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTfBSz4FOMBqjZQXIYLpXD9Uj7Wuhf7m_fWyw&usqp=CAU
● 323
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcQkuQ-EO0e-UWPxscQmnGKPeZePm0nj2TSdWw&usqp=CAU
● 324
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcSmGGMC8oxGRC-h0fw6jO5rgw5_E8UH1TmHBg&usqp=CAU
● 325
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcQ3yAMkT3g5PgYZ-BKHD7BVx7YH8YOQ9AO0qA&usqp=CAU
● 326
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTO5nAQcZInD5sf6d2YK8oONbgugTOkMts3JA&usqp=CAU
● 327
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcSjMOIk3Ajw3kYvPSKZfJYkUtCNthailijiLA&usqp=CAU
● 328
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTfx4h2dF5xO8mUWziFOD3jp4UYgh2I4V5c6w&usqp=CAU
● 329
https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTNO8Wj_gf9eN3UWQTJ2OGs8vv23HICfusQPw&usqp=CAU

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

```
index
```

```
330
```

```
cnt
```

```
120
```

```
#cnt 변수를 120으로 설정했다면...
#final_img_src 리스트에 cnt (120)개의 이미지를 저장할 계획

file_no = 1
for i in range(0,cnt) : #i는 0부터 (cnt-1)119까지 120번 반복
        try :
                urllib.request.urlretrieve(final_img_src[i], str(file_no)+'.jpg')  #이미지 파일명을 file_no(1,2,3...)을 사용
                print("final_img_src 리스트의 %d번 인덱스를 접근하여 '%s.jpg' 로 저장하였습니다. " %(i, file_no))
        except TypeError:
                print("final_img_src 리스트의 %d번 인덱스를 접근하였으나 None 상태입니다." %(i)) #혹시 모를 에러에 대비
                continue

        time.sleep(1)
        print('-'*80)
        file_no += 1
```

```
final_img_src 리스트의 0번 인덱스를 접근하여 '1.jpg' 로 저장하였습니다.
--------------------------------------------------------------------------------
final_img_src 리스트의 1번 인덱스를 접근하여 '2.jpg' 로 저장하였습니다.
--------------------------------------------------------------------------------
final_img_src 리스트의 2번 인덱스를 접근하여 '3.jpg' 로 저장하였습니다
```

이미지 데이터 수집 후 저장(구글 활용)

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise

Step 7. 이미지 저장 후 정보 출력하기

```
e_time = time.time( )
t_time = e_time - s_time

print("구글에서 다음과 같이 이미지가 수집되었습니다.")
print("=" *70)
print("이미지 수집에 사용된 키워드 : ", query_txt)
print("이미지 수집 후 저장된 디렉토리 : ", result_dir)
print("수집된 이미지 개수 : ", cnt) #cnt 대신 file_no-1으로 해도 됨

print("=" *70)
```

```
구글에서 다음과 같이 이미지가 수집되었습니다.
======================================================================
이미지 수집에 사용된 키워드 :  샴고양이
이미지 수집 후 저장된 디렉토리 :  D:₩ai₩DATA₩GOOGLE IMAGE 2020-10-09-15-33-58
수집된 이미지 개수 :  120
======================================================================
```

```
driver.close( )
```

이미지 데이터 수집 후 저장(구글 활용)

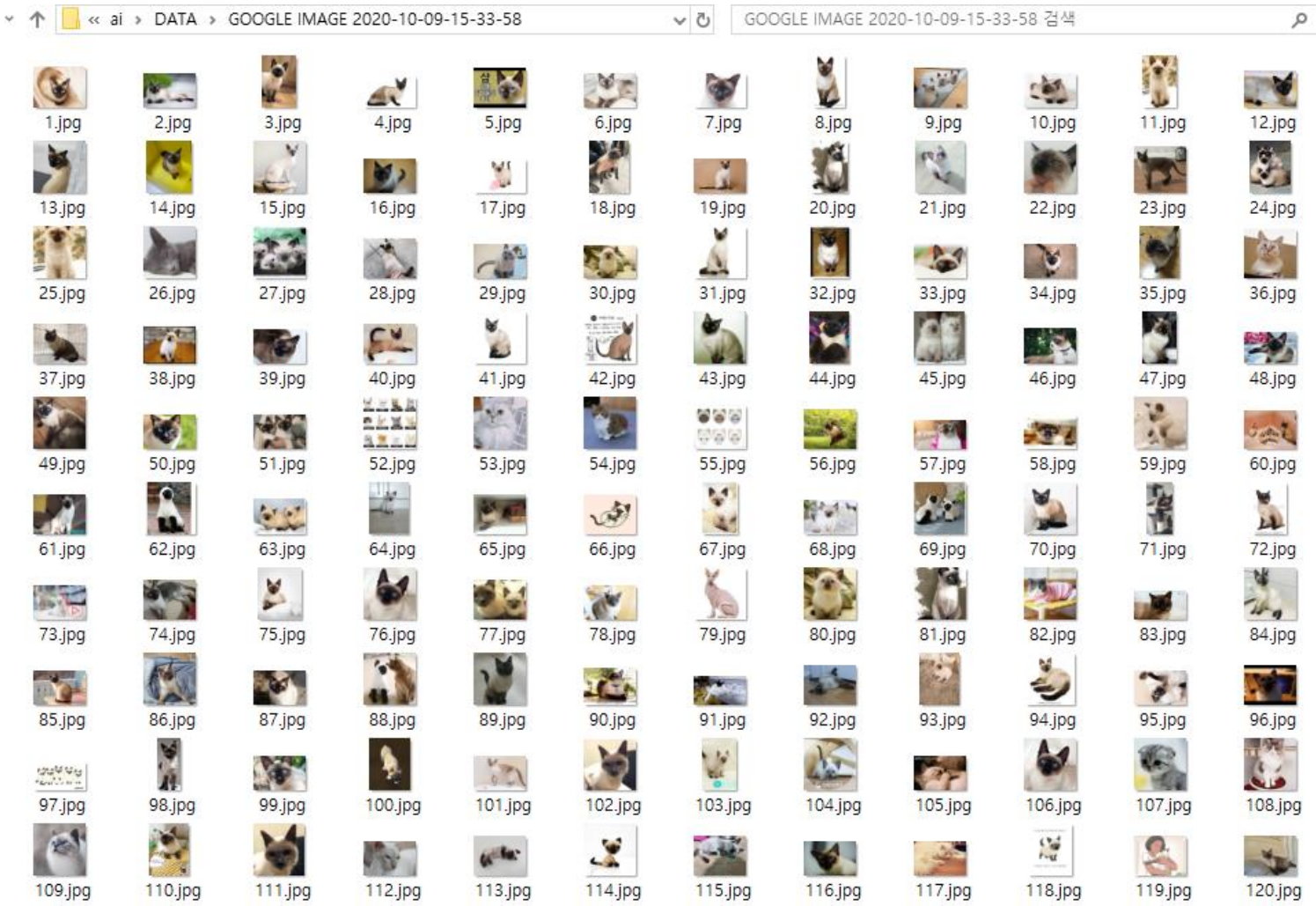
웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- **구글**

- Exercise



Exercise

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글

- Exercise

실습 노트 참고

09~10 Exercise 네이버 블로그 데이터 수집(포함할 키워드, 제외할 키워드,검색 개수,날짜, 정렬 포함).ipynb

네이버(www.naver.com) 에서 검색을 수행하여 블로그 데이터의 내용만 **txt** 형식, **csv** 형식, **xlsx** 형식으로 저장하는 웹 크롤러를 구현하시오.
단, 웹 크롤러를 구현하기 전에 크롤링 결과를 저장할 디렉토리를 아래와 같이 생성한 후, 생성된 디렉토리에 수집된 데이터를 저장하시오.

<웹 크롤링 결과를 저장할 디렉토리>
D:/ai/DATA/Naver_blog_Travel 2021-04-12-17-10-21 여기에서 **2021-04-12-17-10-21**는 오늘날짜 및 시각을 의미합니다.
<웹 크롤러 구현에 포함할 내용>
1.크롤링할 키워드는 무엇입니까**?:** 여행
2.크롤링에 포함할 키워드를 입력하세요**:** 국내
3.크롤링에 제외할 키워드를 입력하세요**:** 해외, 외국
4.크롤링할 데이터 개수를 쓰세요 **:** 100
5.검색을 시작할 날짜를 입력하세요(예**:2019-01-01**)**:** 2020-01-01
6.검색을 종료할 날짜를 입력하세요(예**:2019-12-31**)**:** 2020-12-31
7.txt 형태로 저장할 파일명을 확장자 포함해서 쓰세요**:** Travel.txt
8.csv 형태로 저장할 파일명을 확장자 포함해서 쓰세요**:** Travel.csv
9.xlsx 형태로 저장할 파일명을 확장자 포함해서 쓰세요**:** Travel.xlsx

네이버에서 웹크롤러 구현시 다음과 같은 검색조건을 포함하여 구현하세요.

1단계 **:** 검색어를 입력한다.
2단계 **:** 검색된 데이터 중에서 블로그를 선택한다.
3단계 **:** 검색 옵션을 클릭한다.
3.1 단계 **:** 정렬 메뉴에서 최신순을 선택한다.
3.2 단계 **:** 기간 메뉴에서 직접입력을 선택하여 검색기간 시작일과 검색기간 종료일을 입력한다.
3.3 단계 **:** 상세 검색 메뉴에서 반드시 포함하는 단어에 포함할 키워드를 할당하고, 제외하는 단어에 제외할 키워드를 할당한다.

Exercise

웹에서 이미지 크롤링 후 jpg 형식으로 저장

- 한국관광공사

1. 현재 날짜 및 시간 획득
2. 현재 날짜 및 시간을 디렉토리명에 활용
3. 화면 자동 스크롤
4. 이미지 수집

- 구글

- Exercise

실습 노트 참고

코딩 결과 : pandas 기반의 데이터 프레임 예시

	번호	제목	내용	작성자	작성일
0	1	[국내 맛집 여행] 서울 라멘 맛집 금호동 멘야신, 수준급의...	안녕하세요 세쿨이입니다 :) 오늘은 [국내 맛집 여행 이야기입니다 서울 라멘 맛집 ...	세쿨이의 여행이야기	2021.03.31.
1	2	충남 태안 가볼만한곳 신두리 해안사구 한국관광 100선 선정된...	국내 최대 규모의 사구지역이구요. 천연기념물 431호로 지정되었어요. 신두리... ...	한순간도 소중하게~.	2021.03.31.
2	3	[이슈레포트] 격동의 전기차 시장	자 이제 전기차 시장으로 여행을 떠나 보실까요~? 우선, 왜 전기차 시장이... 이...	상해연합마케팅동아리	2021.03.31.
3	4	신용카드, 삼성 아멕스 플래티늄 메탈 카드 출시, 기존과 다름	무료 여행만 갈 수 있다면 최강이라 생각됩니다. 외국이 싫다면 국내도 1+1 입니다...	JS 지구생활 이야기...	2021.03.31.
4	5	베스킨라빈스31 창업 양도양수 알아보기~!	기술로 국내 최초아이스크림케이크를 만들어 세상에 선보이게 됩니다. 그 후... 한국...	창업따라	2021.03.31.
...	...	...	...	...	...
95	96	[국내섬여행] 파랑새가 사는섬 어청도에서 어청도등대를...	어청도 도보여행 3코스 능선길을 걸어서 어청도등대까지 걸어서 갑니다.... 샘넘길로...	칸의 여행	2021.03.31.
96	97	아름다웠던 우리의 허니문, Jeju	제주 신혼여행 여행지, 여행일정 공유 우리의 허니문은 " 하와이 "로 가겠노라 했지...	Simple Couple	2021.03.31.
97	98	렉스턴스포츠 중고 꼼꼼하게 확인후에	렉스턴스포츠 중고는 국내 SUV 중 프라임바디가 장착되어 있는 차량이에요.... 천...	\ ^ _ ^ /	2021.03.31.
98	99	국내신혼여행 - 군산여행	늦은 신혼여행 (유치원손생님의슬픔) 1일차는 군산! 제일 먼저 지린성으로 갔어요 돌...	팡복이네 신혼일기	2021.03.31.
99	100	지파시 젤라또 엑설런트를 외칠 수 있는 G.FASSI	사람으로써 국내에 들어와 있는 젤라또 브랜드중에 단연 손꼽는 브랜드라고 감히... ...	또젤라또 스토리	2021.03.31.

100 rows × 5 columns