Chap 6. 다양한 이미지 수집 크롤러 만들기

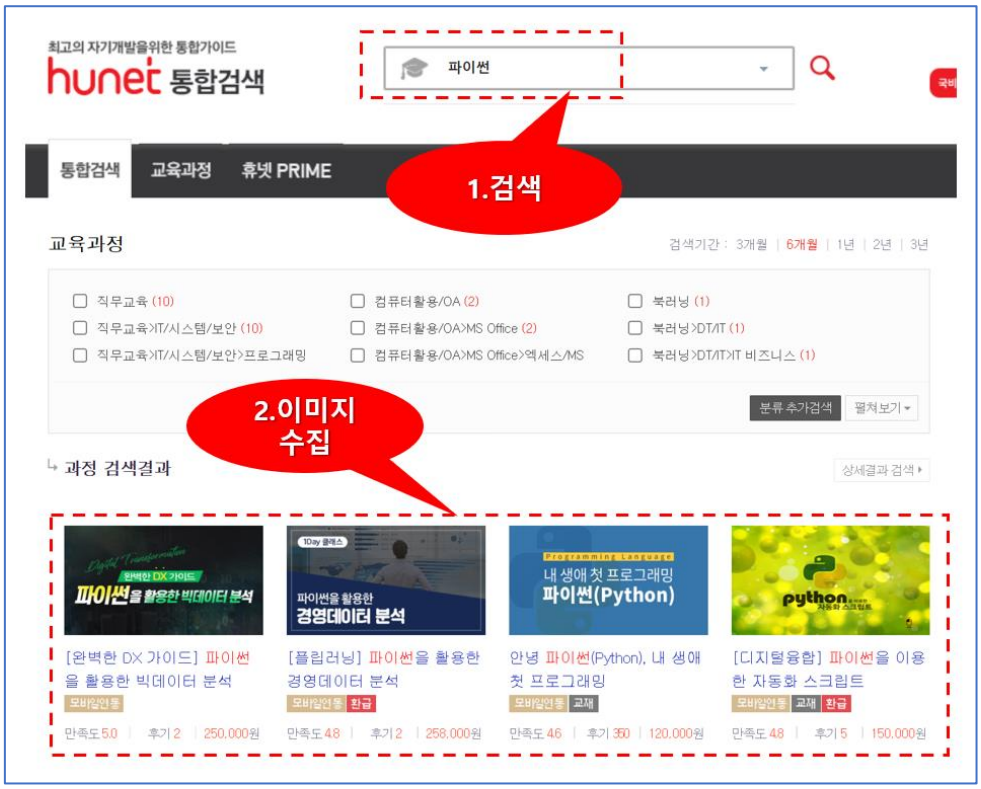
1. 이번 장에서 배울 내용 소개

우리가 지난 시간까지 수집한 데이터들은 전부 텍스트 데이터 위주였습니다.
실제 데이터 수집을 하다 보면 텍스트 유형의 데이터 외에도 그림이나 동영상 등의 데이터를 수집해야 할 경우도 많이 있어서 이번 시간에는 다양한 이미지를 수집하는 내용을 학습합니다.

이미지를 수집하기 위해서는 웹 페이지에서 원하는 이미지의 URL 주소를 추출한 후 웹서버에서 이미지를 다운로드 받는데 이 때 두 가지 경우가 있습니다.
첫 번째는 이미지가 저장된 서버에서 이미지를 가져올 때 인증 정보 없이 그냥 가져올 수 있는 방법이고 두 번째는 인증 정보를 보낸 후 이미지를 가져올 수 있는 경우입니다.

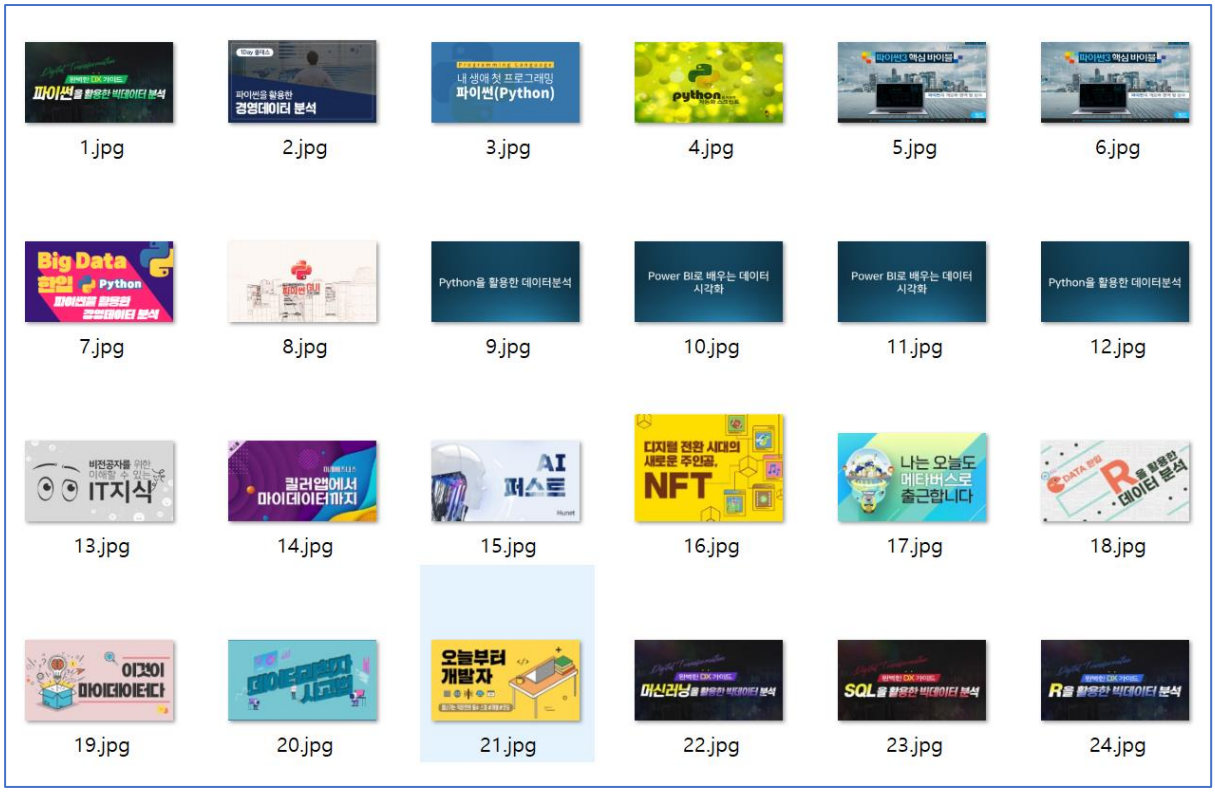
그림으로 이번 챕터에서 배울 내용을 먼저 살펴보겠습니다.
첫 번째 예제는 이미지를 다운로드 받을 때 별도의 과정이 필요 없는 사이트의 경우입니다.
이 예제는 온라인 강의 분야의 최강자이자 다양한 오프라인 강의도 제공하고 있는 휴넷 홈페이지에서 파이썬 강의를 조회하여 강의 목록에 있는 이미지들을 모두 다운로드 받아 저장하는 것입니다.

휴넷 홈페이지 URL : https://www.hunet.co.kr



[파이썬 능력자 너도 될 수 있어~! - 서진수 저 -]

수집한 이미지는 아래와 같이 컴퓨터의 특정 폴더에 자동으로 저장됩니다.



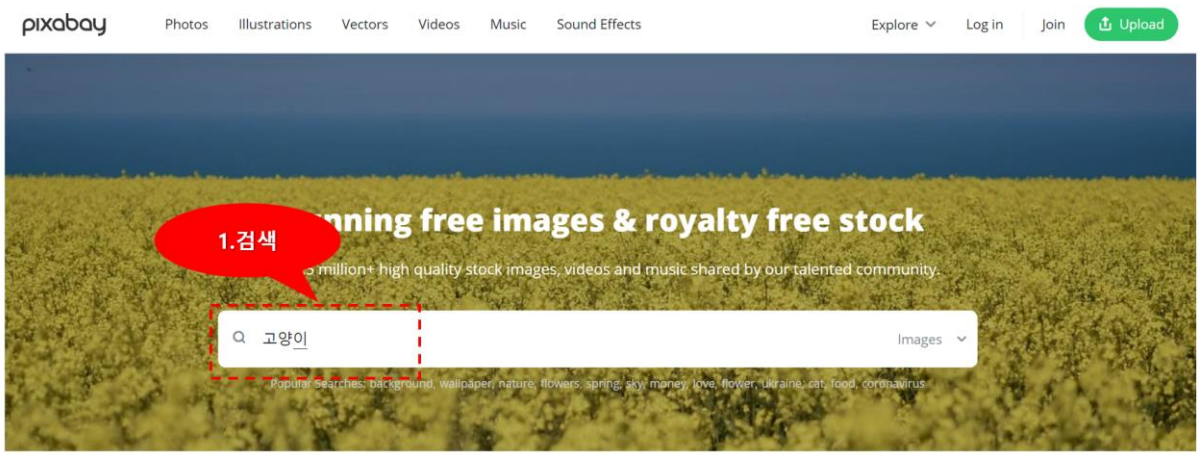
첫 번째 예제에서 배우는 내용 중에서 아주 중요한 부분이 몇가지가 있습니다.

1. 소스코드에서 이미지 파일의 원본 주소를 추출하기
2. 추출된 내용을 다운로드 하는 과정에서 생기는 크롬 경고창을 해결하기
3. 원본 이미지의 경로와 이름에 한글이 포함되어 있을 경우 발생하는 문제 해결하기

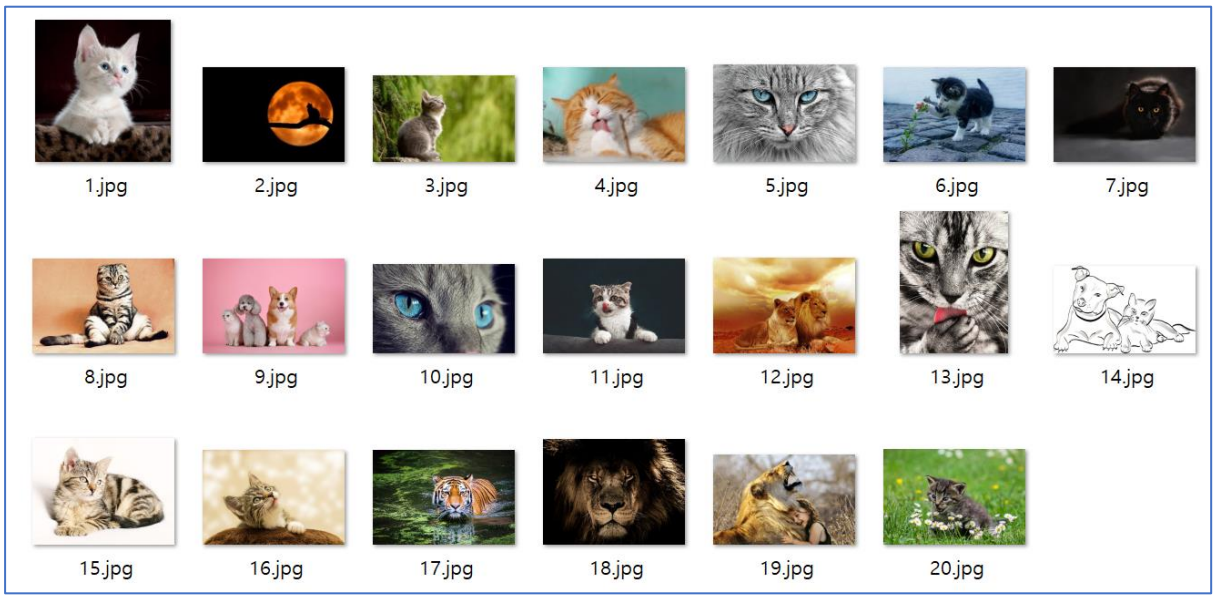
위 3가지 주의사항들을 기억하면서 이후 내용을 봐주세요~

두 번째 예제는 이미지를 다운로드 받을 때 클라이언트의 정보를 제공하는 사이트입니다
이 경우의 예제는 무료 이미지를 다운로드 받을 수 있는 사이트인 pixabay 사이트에 접속하여 "고양이" 키워드로 이미지를 검색한 후 다운로드 받는 예제입니다.

pixabay 홈페이지 URL : https://pixabay.com/



수집된 결과는 아래 그림과 같이 컴퓨터에 파일로 저장이 됩니다.



두 번째 예제를 공부하면서 주의할 부분은 요청을 받은 이미지 서버에서 클라이언트의 정보를 요청하는 경우 어떻게 정보를 주고 데이터를 받아오는가 하는 방법입니다.
이런 경우도 많기 때문에 이 부분을 꼭 기억해 주세요.

2. 전체 소스코드 미리 보기

(아래 소스코드는 저자가 제공해드리는 소스코드를 다운로드 받아 사용하세요)

[Case 1 – 클라이언트의 정보가 필요 없는 경우]

```
# Case 1 – 클라이언트 정보가 필요 없는 경우
# Step 1. 필요한 모듈과 라이브러리를 로딩하고 검색어를 입력 받습니다
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.service import Service
from bs4 import BeautifulSoup
import urllib.request
import urllib.parse
import time
import math
import os
import random

#Step 2. 사용자에게 검색 관련 정보들을 입력 받습니다.
print("=" *100)
print(" 이 크롤러는 휴넷 사이트의 강의 자료 수집용 웹크롤러입니다.")
print("=" *100)
query_txt = input('1.수집할 자료의 키워드는 무엇입니까?(예: 파이썬): ')

try :
    cnt = int( input('2.수집할 건수는 총 몇건입니까?(기본값:10): '))
except ValueError :
    cnt = 10
    print('기본값인 10 건으로 수집을 진행합니다.')
page_cnt = math.ceil( cnt / 12)

f_dir=input('3.파일이 저장될 경로만 쓰세요(예: c:₩₩py_temp₩₩ ) : ')
if f_dir =='' :
    f_dir = "c:₩₩py_temp₩₩"

#Step 3. 크롬 드라이버 설정 및 웹 페이지 열기
s = Service("c:/py_temp/chromedriver.exe")
driver = webdriver.Chrome(service=s)

```

```
36  url = 'https://www.hunet.co.kr/'
37  driver.get(url)
38  driver.maximize_window()
39  time.sleep(3)
40
41  #Step 4. 자동으로 검색어 입력 후 조회하기
42  element = driver.find_element(By.ID,'txtKeyword')
43  driver.find_element(By.ID,'txtKeyword').click( )
44  element.send_keys(query_txt)
45  element.send_keys("₩n")
46
47  #Step 5. 사용자가 요청한 건수만큼 더보기 클릭하기
48  for a in range(0,page_cnt) :
49      try :
50          driver.find_element(By.XPATH,'//*[@id="divEducationList"]/div/div[2]/div[5]/a[1]').click()
51          time.sleep(2)
52          try :
53              result = driver.switch_to_alert()
54              result.accept( )
55          except :
56              continue
57      except :
58          print('페이지 이동이 끝났습니다. 이제 데이터를 수집하겠습니다')
59          break
60
61  # Step 6. 이미지 추출하여 저장하기
62  file_no = 0
63  count = 1
64  img_src2=[]  # 이미지 원본 URL 주소 저장할 리스트
65
66  n = time.localtime()
67  s = '%04d-%02d-%02d-%02d-%02d-%02d' % (n.tm_year, n.tm_mon, n.tm_mday, n.tm_hour, n.tm_min, n.tm_sec)
68  img_dir = f_dir+s+'-'+query_txt
69  os.makedirs(img_dir)
70  os.chdir(img_dir)
71
72  html = driver.page_source
73  soup = BeautifulSoup(html, 'html.parser')
74  img_src = soup.find('ul','vod_list').find_all('img')
```

```
75
76  for i in img_src :
77          img_src1=i['src']
78          img_src2.append(img_src1)
79          print(img_src1)
80          count += 1
81          if count > cnt :
82                  break
83
84  for i in range(0,len(img_src2)) :
85          file_no += 1
86          #이미지 파일에 한글 이름이 들어갈 경우 오류가 발생하므로 인코딩 지정해 주어야 함
87          urllib.request.urlretrieve(urllib.parse.quote(img_src2[i].encode('utf8'), '/:'),str(file_no)+'.jpg')
88
89          time.sleep(0.5)
90          print("%s 번째 이미지 저장중입니다=======" %file_no)
91
92  # Step 7. 요약 정보를 출력합니다
93  print("=" *70)
94  print("총 저장 건수는 %s 건 입니다 " %file_no)
95  print("파일 저장 경로: %s 입니다" %img_dir)
96  print("=" *70)
97  driver.close( )
**************************************************************************************
```

[Case 2 – 클라이언트 정보가 필요한 경우]

```
# Case 2. pixapay 사이트에서 그림 수집하기
# Step 1. 필요한 모듈과 라이브러리를 로딩합니다.

from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.service import Service
import urllib.request
import urllib
import time
import math
import os
import random

#Step 2. 필요한 정보를 입력 받습니다.
print("=" *80)
print(" pixabay 사이트에서 이미지를 검색하여 수집하는 크롤러 입니다 ")
print("=" *80)

query_txt = input('1.크롤링할 이미지의 키워드는 무엇입니까?: ')
cnt = int(input('2.크롤링 할 건수는 몇건입니까?: '))
real_cnt = math.ceil(cnt / 100) # 실제 크롤링 할 페이지 수
f_dir=input('3.파일이 저장될 경로만 쓰세요(예: c:₩₩py_temp₩₩ ) : ')
if f_dir =='' :
    f_dir = "c:₩₩py_temp₩₩"

print("₩n")
print("요청하신 데이터를 수집 중이오니 잠시만 기다려 주세요~~^^")

#Step 3. 파일을 저장할 폴더를 생성합니다
n = time.localtime()
s = '%04d-%02d-%02d-%02d-%02d-%02d' % (n.tm_year, n.tm_mon, n.tm_mday, n.tm_hour, n.tm_min, n.tm_sec)

img_dir = f_dir+s+'-'+query_txt
os.makedirs(img_dir)
os.chdir(img_dir)
```

```
38
39  #Step 4. 크롬 드라이버를 사용해서 웹 브라우저를 실행한 후 검색합니다
40  s_time = time.time( )
41
42  s = Service("c:/py_temp/chromedriver.exe")
43  driver = webdriver.Chrome(service=s)
44
45  driver.get('https://pixabay.com/ko/')
46  time.sleep(3)
47
48  # 검색어 입력 창에 검색어 입력 후 검색 수행
49  element = driver.find_element(By.NAME,'q')
50  element.send_keys(query_txt)
51  element.submit()
52
53  # Step 5. 이미지 추출하여 저장하기
54  file_no = 1
55  count = 1
56  img_src2=[]      #이미지 파일의 url 주소 저장용 리스트
57
58  # 스크롤 다운 함수 만들기
59  def scroll_down(driver):
60      #driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
61      driver.execute_script("window.scrollBy(0,1000)")
62      time.sleep(1)
63
64  for a in range(1 , real_cnt+1):
65
66      for b in range(0,5):
67          scroll_down(driver)
68          time.sleep(1)
69
70      # 원본 이미지 url 주소 수집
71      html = driver.page_source
72      soup = BeautifulSoup(html, 'html.parser')
73      img_src = soup.find('div','container--HcTw2').find_all('img')
74
75      for c in img_src :
76          img_src1=c['src']
```

```
77
78              if 'http' in img_src1 :
79                  img_src2.append(img_src1)
80                  print(img_src1)
81
82              count += 1
83
84              if count > cnt :
85                  break
86
87          #수집된 url 주소로 이미지 파일 가져와서 저장하기
88          for e in range(0,len(img_src2)) :
89
90            class AppURLopener(urllib.request.FancyURLopener):
91                  version = "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, ₩
92                  like Gecko) Chrome/47.0.2526.69 Safari/537.36"
93
94              urllib.urlopener = AppURLopener()
95              urllib.urlopener.retrieve(img_src2[e],str(file_no)+'.jpg')
96
97              print("%s 페이지에서 %s 번째 이미지 저장중입니다=======" %(a,file_no))
98
99              time.sleep(0.5)
100
101             if file_no >= cnt :
102                     break
103
104             file_no += 1
105
106         if a > real_cnt :
107             break
108
109 # Step 6. 요약 정보를 출력합니다
110 e_time = time.time( )
111 t_time = e_time - s_time
112
113 print("=" *70)
114 print("총 소요시간은 %s 초 입니다 " %round(t_time,1))
115 print("총 저장 건수는 %s 건 입니다 " %file_no)
```

```
116 print("파일 저장 경로: %s 입니다" %img_dir)
117 print("=" *70)
118 driver.close( )
```

****

소스코드가 많이 길죠?

자세하게 설명해 드릴 테니까 설명을 잘 보고 공부해주세요~

3.소스 코드 설명

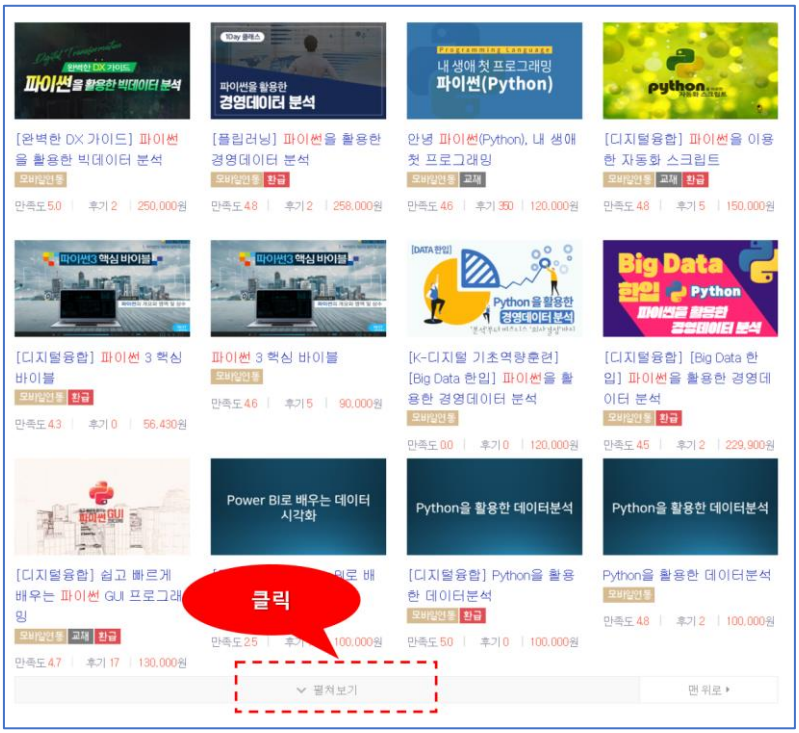
위 코드 중에서 중요한 부분을 정리해서 설명하겠습니다.

[Case 1 – 클라이언트 정보가 필요 없는 경우]

```
47  #Step 5. 사용자가 요청한 건수만큼 더보기 클릭하기
48  for a in range(0,page_cnt) :
49      try :
50          driver.find_element(By.XPATH,'//*[@id="divEducationList"]/div/div[2]/div[5]/a[1]').click()
51          time.sleep(2)
52          try :
53              result = driver.switch_to_alert()
54              result.accept( )
55          except :
56              continue
57      except :
58          print('페이지 이동이 끝났습니다. 이제 데이터를 수집하겠습니다')
59          break
```

위 코드는 휴넷 사이트에서 파이썬으로 검색한 후 사용자가 요청한 건수만큼 스크롤을 내려서 페이지를 이동하는 부분입니다.

휴넷 사이트의 경우 아래 그림과 같이 한 페이지에 12개의 그림이 있고 다음 페이지로 가고 싶을 경우 아래의 **펼쳐보기** 버튼을 클릭해야 합니다.



소스코드의 50번행이 펼쳐보기 버튼을 xpath 값을 사용해서 클릭하는 부분입니다.
그런데 문제는 사용자에게 총 몇 건을 수집 할 것인지 물어봤을 때 사용자가 이미지의 개수보다 많은 값을 입력할 경우입니다.
예를 들어 전체 이미지는 50건 밖에 없는데 사용자가 100건 수집하라고 할 경우가 문제가 되는 거죠. 페이지 번호를 계산할 시점에서는 총 이미지가 몇개 있는 지 알 수가 없을 경우가 많은데 예를 들어 총 50개의 이미지밖에 없는데 사용자가 100건을 요청할 경우 다음 페이지로 넘어가기 위해 펼쳐보기 버튼을 반복해서 누르다가 더이상 페이지가 없을 경우 크롬에서는 경고창을 아래 그림과 같이 출력합니다.
이때 확인 버튼을 클릭한 후 펼쳐보기 버튼 누르는 것을 멈춰야 합니다.

본문의 53-54번 행이 바로 크롬에서 alert 창이 열렸을 경우 포커스를 alert 창으로 바꾼 후 확인 버튼을 클릭하는 부분입니다.
크롬에서 어떤 작업을 하다가 위의 예제와 같이 alert 창이 열리는 경우가 아주 많은데 이런 경우에 아주 요긴하게 사용할 수 있는 방법이니까 잘 기억해 주세요~
그리고 58-59번 행에서 더이상 펼쳐보기 버튼을 클릭하지 말고 종료하도록 코드를 작성했습니다.

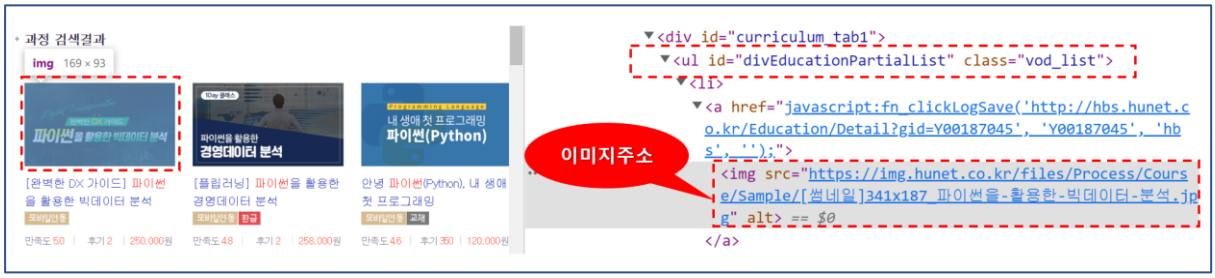
이제 이미지 정보를 저장할 폴더를 생성하는 부분입니다.

```
66  n = time.localtime()
67  s = '%04d-%02d-%02d-%02d-%02d-%02d' % (n.tm_year, n.tm_mon, n.tm_mday, n.tm_hour, n.tm_min, n.tm_sec)
68  img_dir = f_dir+s+'-'+query_txt
69  os.makedirs(img_dir)
70  os.chdir(img_dir)
```

위 코드에서 이미지를 저장할 폴더 이름에 크롤링을 하는 시점의 날짜와 시간 정보를 넣고 폴더를 생성하도록 코드를 작성했습니다.
이제 본문의 소스코드에서 이미지의 URL 주소를 추출하겠습니다.

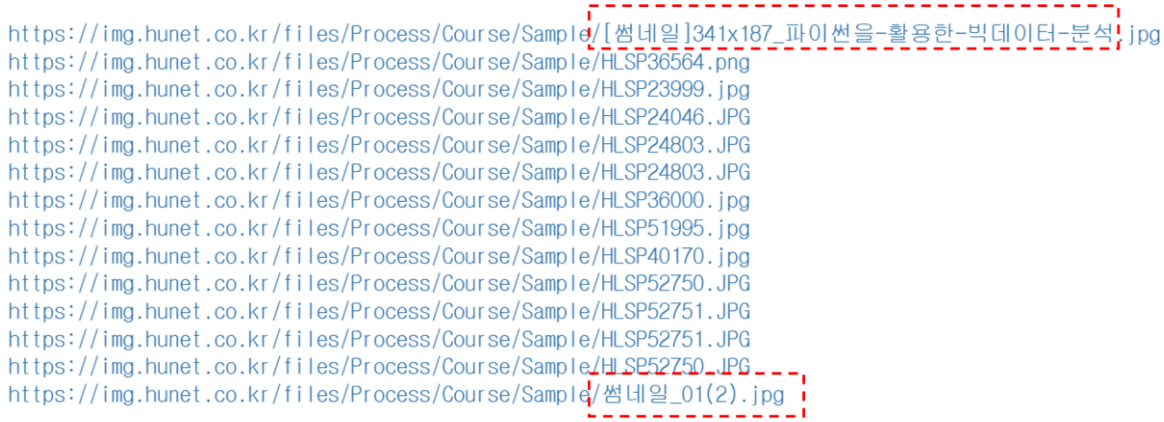
```
72  html = driver.page_source
73  soup = BeautifulSoup(html, 'html.parser')
74  img_src = soup.find('ul','vod_list').find_all('img')
75
76  for i in img_src :
77      img_src1=i['src']
78      img_src2.append(img_src1)
79
80      count += 1
81      if count > cnt :
82          break
```

위 코드의 74번행이 현재 페이지에서 이미지 정보를 저장하고 있는 "img" 태그를 모두 추출하는 코드입니다. 웹 페이지에서 이미지 정보를 저장하고 있는 소스코드를 살펴볼까요?



위 그림을 보면 이미지의 URL 주소가 <ul class="vod_list"> 아래에 <img src> 부분에 있는 것이 보이죠? 그래서 소스코드의 74번 행과 같이 코딩을 했습니다.
74번 행을 수행하면 img_src 변수에는 현재 페이지에 있는 모든 이미지의 img 태그가 추출되겠죠? 그리고 img 태그의 속성 중에서 src 속성값에 URL 주소가 있어서 76번부터 78번행까지 반복문으로 src 속성값을 추출하여 img_src2 리스트에 추가를 합니다.
여기까지 하면 img_src2 리스트에는 모든 이미지의 URL 주소가 들어가게 됩니다.

이제 가장 중요한 작업을 해야 합니다.
바로 이미지를 가져와서 내 컴퓨터의 폴더에 저장하는 단계입니다.
이미지를 가져와서 저장하는 것은 urllib.request.urlretrieve(이미지주소 , 저장할 경로와이름) 함수를 사용하면 쉽게 처리할 수 있습니다.
그런데 이 단계에서 아주 중요한 문제가 생길 수 있습니다.
바로 가져올 이미지 주소에 한글이 들어 있으면 에러가 발생하는 문제입니다.
아래 그림을 보세요.



위 그림의 첫번째 줄과 마지막 줄처럼 이미지 주소에서 한글이 있는 부분 보이죠?
이럴 경우 urllib.request.urlretrieve( ) 함수가 한글을 제대로 인식을 못해서 오류가 발생합니다.
그래서 예외처리를 사용하여 건너뛰거나 인코딩을 지정하는 방법으로 이미지 이름을 인식시켜서 다운로드를 받아야 하는데 이 부분의 코드가 아래와 같습니다.

```
84  for i in range(0,len(img_src2)) :
85      file_no += 1
86          #이미지 파일에 한글 이름이 들어갈 경우 오류가 발생하므로 인코딩 지정해 주어야 함
87          urllib.request.urlretrieve(urllib.parse.quote(img_src2[i].encode('utf8'), '/:'),str(file_no)+'.jpg')
88
89      time.sleep(0.5)
90      print("%s 번째 이미지 저장중입니다=======" %file_no)
```

위 코드에서 87번 행이 이미지를 다운로드해서 내 컴퓨터에 저장하는 부분입니다.
그런데 만약 이미지 이름에 한글이 들어갈 경우 문제가 생기기 때문에 87번행과 같이 encode( ) 함수로 인코딩을 지정해 주어야 합니다.
이미지 파일 이름에 한글이 있는 경우가 아주 많기 때문에 이 부분을 잘 활용하세요~

[Case 2 – 클라이언트 정보가 필요한 경우]

앞의 Case 1과 중복되는 부분은 설명을 생략하고 이번 예제에서 중요한 부분만 설명을 하겠습니다.

먼저 pixabay 사이트의 경우 많은 이미지를 보려면 페이지 버튼을 클릭하는 대신 화면을 아래로 스크롤 다운을 해야 추가 이미지들이 보여집니다.

그래서 아래와 같이 화면을 아래로 내리는 사용자 정의 함수를 생성한 후 실행해서 화면을 아래로 이동한 후 소스코드를 가져와야 합니다.

```
58  # 스크롤 다운 함수 만들기
59  def scroll_down(driver):
60          #driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
61          driver.execute_script("window.scrollBy(0,1000)")
62          time.sleep(1)
63
64  for a in range(1 , real_cnt+1):
65
66          for b in range(0,5):
67                  scroll_down(driver)
68                  time.sleep(1)
```

위 코드의 59번 행부터 62번 행은 화면을 자동으로 스크롤 다운해서 이동시켜주는 함수입니다.

driver.execute_script( ) 함수를 사용한 것은 파이썬 코드에서 외부 OS 에 있는 특정 함수나 스크립트를 실행할 때 많이 사용하는 방법입니다. 즉 이 함수의 괄호안에 우리가 실행하고 싶은 OS의 함수나 기능을 적으면 되는데 우리는 윈도의 마우스 스크롤 하는 기능을 실행하기 위해서 window.scrollTo( ) 함수를 사용한 것입니다.

window.scrollTo(x좌표, y좌표) 형식으로 사용하는데 예를 들어 window.scrollTo(0, 500) 이라고 적으면 500 픽셀만큼 아래로 화면을 이동시켜 줍니다.

이 함수와 비슷한 함수로 window.scrollBy(x좌표 , y좌표) 의 함수도 있는데 의미는 동일하지만 차이점은 window.scrollTo( ) 함수는 기준값이 절대 좌표이고 window.scrollBy( ) 함수는 기준값이 상대좌표입니다. 화면 끝까지 이동하고 싶을 경우에는 document.body.scrollHeight 값을 사용하면 됩니다.

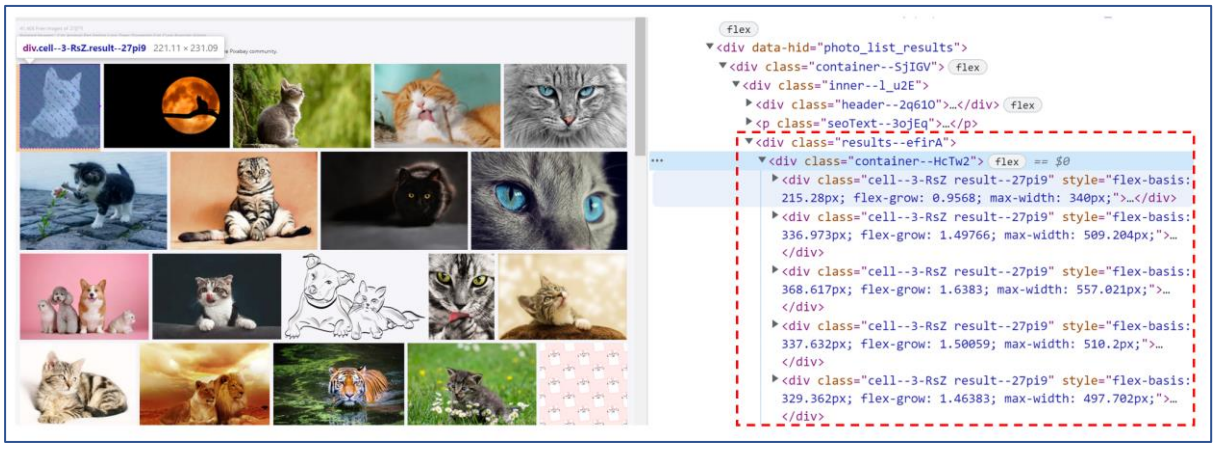
pixabay 사이트에서 많은 이미지를 추출하기 위해서는 화면을 스크롤 다운해서 아래로 이동해야 하기 때문에 이 함수를 생성한 후 64번 행부터 68번 행까지 이 함수를 실행하여 화면을 아래로 스크롤 다운시켰습니다.

이제 본격적으로 이미지들의 URL 주소값을 추출해야 합니다.

아래 코드를 보세요.

```
70        # 원본 이미지 url 주소 수집
71        html = driver.page_source
72        soup = BeautifulSoup(html, 'html.parser')
73        img_src = soup.find('div','container--HcTw2').find_all('img')
74
75        for c in img_src :
76            img_src1=c['src']
77
78            if 'http' in img_src1 :
79                img_src2.append(img_src1)
80                print(img_src1)
81
82            count += 1
83
84            if count > cnt :
85                break
86
```

위 코드의 73번 행에서 img 태그 값을 모두 추출하는데 아래 그림을 보면 <div class=" container--HcTw2" 아래에 이미지들이 모두 들어 있는 것이 확인됩니다.



그래서 위 코드의 73번 행과같이 코딩을 했습니다.
그 후 75번 행부터 src 속성값을 추출하여 이미지 정보를 img_src2 리스트에 추가하였습니다.
이미지의 URL 정보를 추출했으니 이제 이미지를 가져와서 저장해야겠죠?
그런데 문제는 이미지를 가지고 있는 pixabay 서버가 클라이언트의 정보를 요청하기 때문에 앞에서 살펴본 방법으로는 사용할 수 가 없습니다.
다음 코드를 보세요.

```
87        #수집된 url 주소로 이미지 파일 가져와서 저장하기
88        for e in range(0,len(img_src2)) :
89
90            class AppURLopener(urllib.request.FancyURLopener):
91                    version = "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, ₩
92                    like Gecko) Chrome/47.0.2526.69 Safari/537.36"
93
94                urllib.urlopener = AppURLopener()
95                urllib.urlopener.retrieve(img_src2[e],str(file_no)+'.jpg')
96
97                print("%s 페이지에서 %s 번째 이미지 저장중입니다=======" %(a,file_no))
98
```

위 코드에서 90번 행부터 92번 행까지가 크롬 드라이버의 정보를 지정하는 부분입니다.
크롬 웹드라이버를 사용할 경우 위와 같이 브라우저의 정보를 서버쪽에 알려주도록 클래스를 만들고 94 , 95번 행에서 서버에 접속 한 후 이미지를 다운로드 받아서 내컴퓨터에 저장하면 됩니다.
참고로 91,92 번 행에 적는 내용은 웹 드라이버의 종류에 따라 다른 내용이 사용되고 위 내용은 크롬을 기준으로 작성한 내용입니다.

이번 시간에는 다양한 이미지를 다운로드 받는 방법을 학습했습니다.
열심히 연습해서 꼭 여러분들의 실력으로 만드세요~~

4. 연습 문제로 실력 굳히기

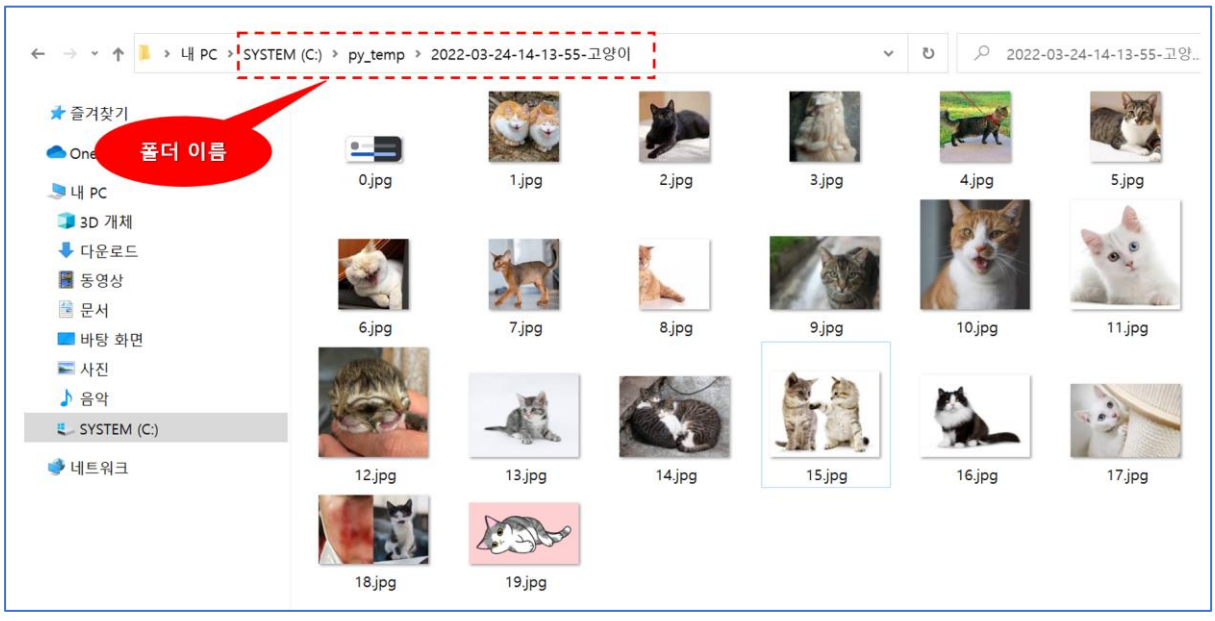
1. 구글에서 특정 키워드로 이미지를 검색해서 컴퓨터의 지정된 폴더로 다운로드하는 크롤러를 만드세요. 이때 저장할 폴더명은 크롤링을 수행하는 현재의 날짜와 시간 값을 포함해 생성하세요.

[크롤러 실행화면]

```
================================================================================
구글 사이트에서 이미지를 검색하여 수집하는 크롤러 입니다
================================================================================
1.크롤링할 이미지의 키워드는 무엇입니까?: 고양이
2.크롤링 할 건수는 몇건입니까?: 20
3.파일이 저장될 경로만 쓰세요(예: c:₩py_temp₩ ) : c:₩py_temp₩
```

위와 같이 검색할 키워드 / 수집 건수 / 저장할 폴더명을 입력 받은 후 구글사이트에서 해당 검색어로 사진을 수집하여 아래 그림처럼 저장하면 됩니다.

[수집된 사진 결과]



2. 네이버 사이트에서 특정 키워드로 이미지를 검색한 후 컴퓨터의 지정된 폴더로 다운로드하는 크롤러를 만드세요.

아래 그림과 같이 크롤링할 이미지 키워드 / 크롤링 건수 / 파일이 저장될 폴더명을 입력 받아서 크롤링을 진행하세요. 단 파일을 저장할 폴더명에 크롤링을 진행하는 현재 날짜와 시간을 포함하여 폴더를 생성하세요.

```
================================================================================
 이 크롤러는 네이버 이미지 정보를 수집합니다
================================================================================
1.크롤링할 이미지의 키워드는 무엇입니까?: 강아지
2.크롤링 할 건수는 몇건입니까?: 25
3.파일이 저장될 경로만 쓰세요(예: c:₩py_temp₩ ) : c:₩py_temp₩
```

[수집된 사진 결과]