

Chap 1. 웹 크롤링을 위한 환경설정과 자동 검색 구현

국민 여러분~^^

웹 크롤링의 세계에 오신 것을 환영합니다~

지금 이 책을 보고 계신 이유는 인터넷에 있는 다양한 데이터를 수집하기 위해서죠?

제가 지금부터 자세하게 전해 드릴 테니까 차근차근 잘 배워서 꼭 능력자가 되세요~~^^

그리고 미리 당부를 드릴 내용이 있어요.

이번 챕터부터 하는 내용들은 파이썬 기본 문법들을 잘 알고 있다고 전제하고 진행을 할거예요.

그래서 혹시라도 파이썬의 기본문법이 잘 기억이 안 나는 분들은 기본 문법 파트를 참고하면서 보면 훨씬 이해가 잘 되실 거예요. 기본 문법이 이해가 안된다고 좌절하지 말라는 의미입니다!

자.....그러면...우리가 이번 챕터에서 배워야 할 목표를 먼저 정리해 볼까요?

**** 이번 챕터의 학습 목표 ****

1. 웹 크롤링의 원리를 이해하고 설명할 수 있다.
2. 웹 크롤링을 위한 주요 모듈들을 설치하고 환경설정을 할 수 있다.
3. 웹사이트에서 검색 자동화를 구현할 수 있다.

위의 학습목표를 모두 공부하면 여러분들은 다양한 사이트에서 검색의 도사가 될 수 있습니다~

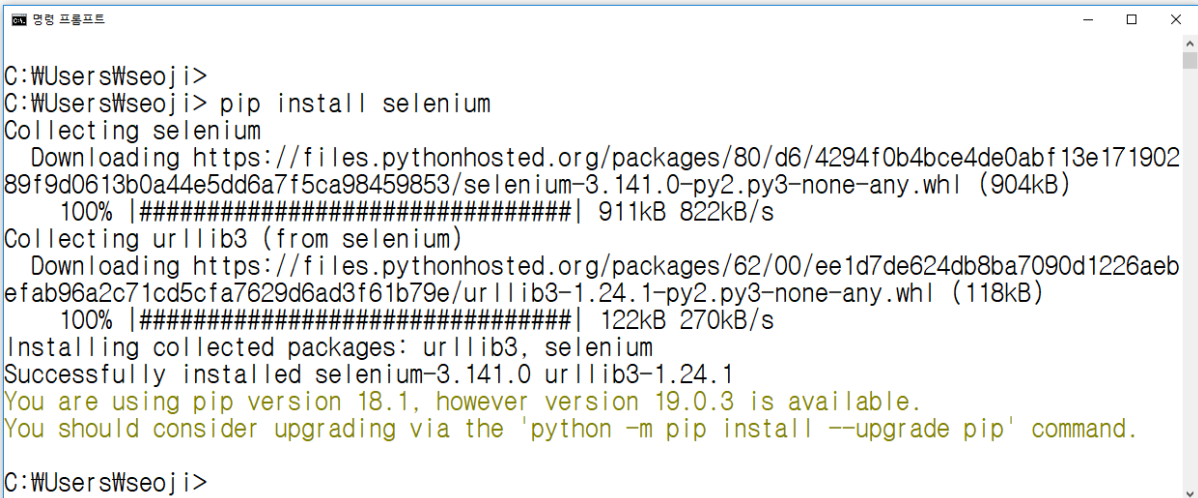
그럼 이제 본격적으로 시작해 볼까요?

2. 웹 크롤링을 위한 설정

이번 챕터에서는 자동 검색 기능을 구현하기 위한 설정을 하고 다음 챕터에서 Beautiful Soup를 설치하고 살펴 볼 예정입니다. 아래 순서대로 진행해 주세요.

1) 사람 대신 웹 페이지를 열고 데이터를 수집할 selenium 패키지를 설치합니다.

웹 크롤러를 만드는 다양한 방법이 있는데 이 책에서는 아주 쉽게 작업할 수 있는 selenium (셀레니움) 이라는 패키지를 사용합니다. selenium 패키지는 사람을 대신해서 주어진 작업을 해 주는 조수 프로그램이라고 생각하면 됩니다. 아래와 같이 윈도의 명령 프롬프트 창에서 "pip install selenium" 을 실행하면 자동으로 다운로드 받은 후 설치가 됩니다.



```

C:\Users\Wseoji>
C:\Users\Wseoji> pip install selenium
Collecting selenium
  Downloading https://files.pythonhosted.org/packages/80/d6/4294f0b4bce4de0abf13e17190289f9d0613b0a44e5dd6a7f5ca98459853/selenium-3.141.0-py2.py3-none-any.whl (904kB)
    100% |#####| 911kB 822kB/s
Collecting urllib3 (from selenium)
  Downloading https://files.pythonhosted.org/packages/62/00/ee1d7de624db8ba7090d1226aebefab96a2c71cd5cfa7629d6ad3f61b79e/urllib3-1.24.1-py2.py3-none-any.whl (118kB)
    100% |#####| 122kB 270kB/s
Installing collected packages: urllib3, selenium
Successfully installed selenium-3.141.0 urllib3-1.24.1
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\Wseoji>
  
```

2) selenium 패키지가 사용할 웹 브라우저 프로그램(웹드라이버)을 설치합니다

웹 데이터를 크롤링 할 때 사람 대신 selenium이 우리가 지금 설치하는 웹 드라이버 프로그램을 이용해서 웹 페이지를 열고 사람이 지정한 태그를 찾아서 데이터를 수집합니다.

사람이 인터넷을 할 때 웹 브라우저로 인터넷 익스플로러나 크롬, 사파리, Edge 등의 프로그램을 사용하는 것과 동일한 원리입니다. 이 책에서는 웹 드라이버 프로그램으로 google chrome 프로그램을 사용하겠습니다.

꼭 기억해야 할 주의 사항은 chrome 웹 브라우저는 사람이 사용하는 것과 selenium 이 사용하는 프로그램이 다르기 때문에 여러분들의 컴퓨터에 먼저 사람이 사용하는 google chrome 이 설치되어 있어야 합니다. 만약 사람이 사용하는 google chrome 웹 브라우저가 설치되어 있지 않거나 chrome 웹 브라우저와 크롬 웹드라이버가 호환이 되지 않을 경우 웹 크롤링을 진행할 때 "selenium.common.exceptions.WebDriverException: Message: unknown error: cannot find Chrome binary" 라는 오류가 나오면서 크롬 드라이버가 실행이 안되니까 꼭 기억해 주세요~

이제 selenium 이 사용할 크롬을 아래와 같이 다운로드 해주세요.

Step 1. <https://sites.google.com/a/chromium.org/chromedriver/downloads> 페이지를 엽니다.

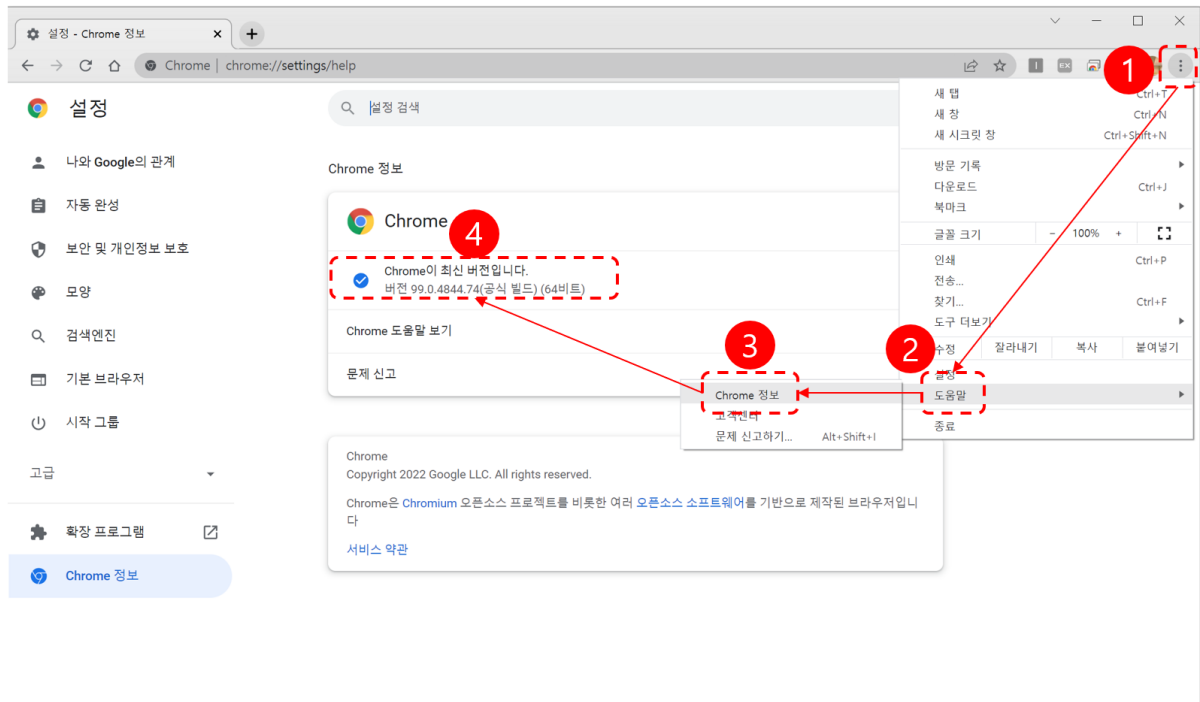
(위 URL을 모두 입력하기 어려우면 www.google.co.kr 에서 chromedriver download 로 검색하면 다운로드 받는 링크가 나옵니다)

위 링크에 접속하면 chrome Web driver 다운로드 홈페이지에 접속할 수 있습니다.

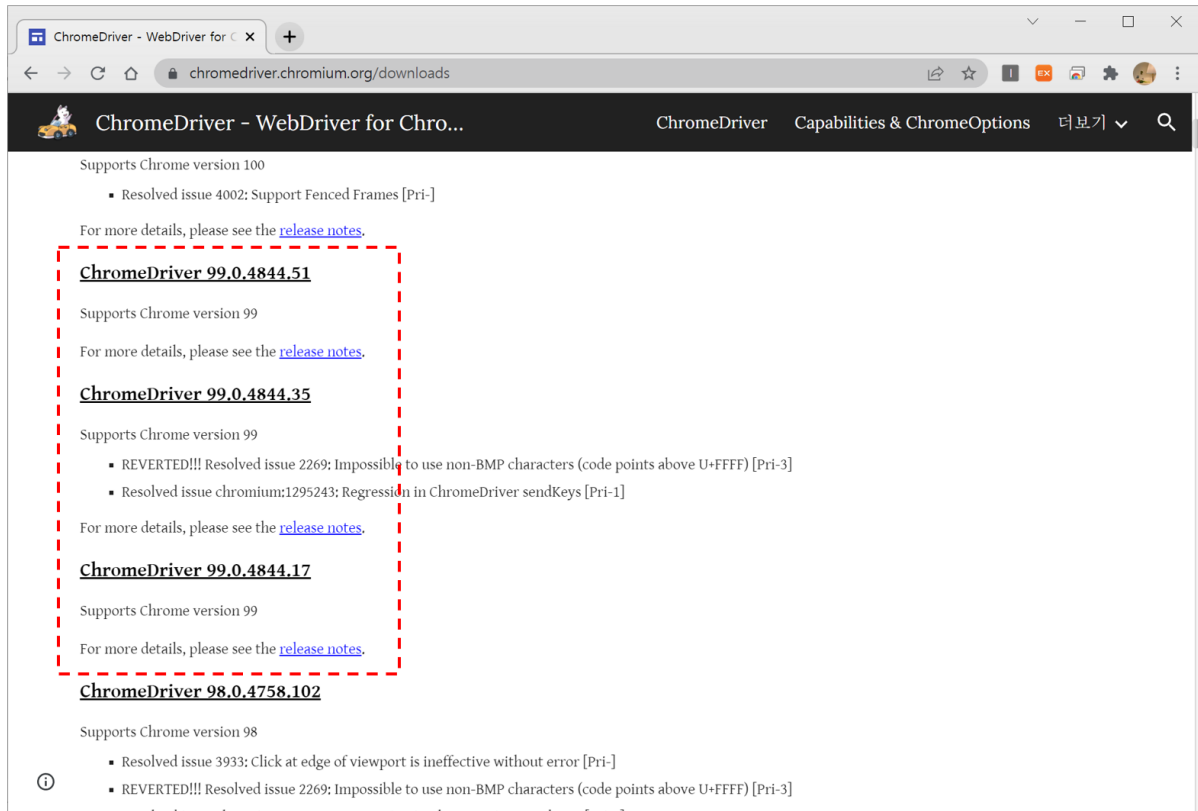
위 사이트에 접속하면 다양한 버전의 크롬 드라이버가 나옵니다. 크롬 드라이버를 선택하는 권장 사항은 현재 컴퓨터에 설치되어 있는 사람이 사용하는 크롬 버전과 동일하거나 비슷한 버전을 설치하는 것입니다.

2022년 3월 10일 기준으로 제가 쓰고 있는 크롬의 버전은 99.0.4844 버전이라서 이 버전과 비슷한 버전의 크롬 드라이버를 다운로드 하겠습니다.

사람이 사용하는 크롬의 버전은 아래와 같이 확인 가능합니다.



크롬 드라이버 다운로드 사이트에 가면 여러가지 버전이 보이는데 그 중에서 아래 그림과 같이 현재 크롬 버전과 동일한 크롬드라이버를 클릭합니다.



비슷한 버전이 여러 개가 나오면 99.0 까지만 동일하면 나머지는 달라도 상관없으니 적당한 버전으로 클릭하세요.

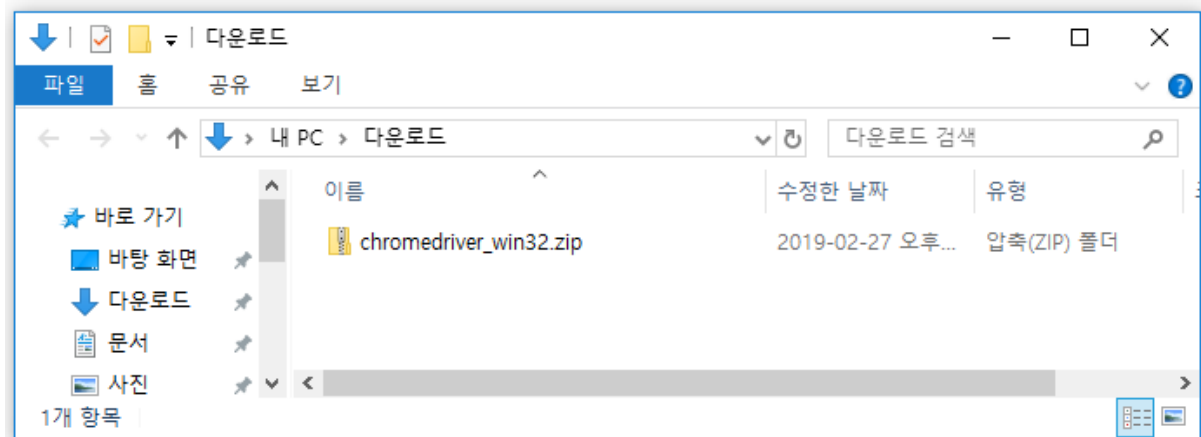
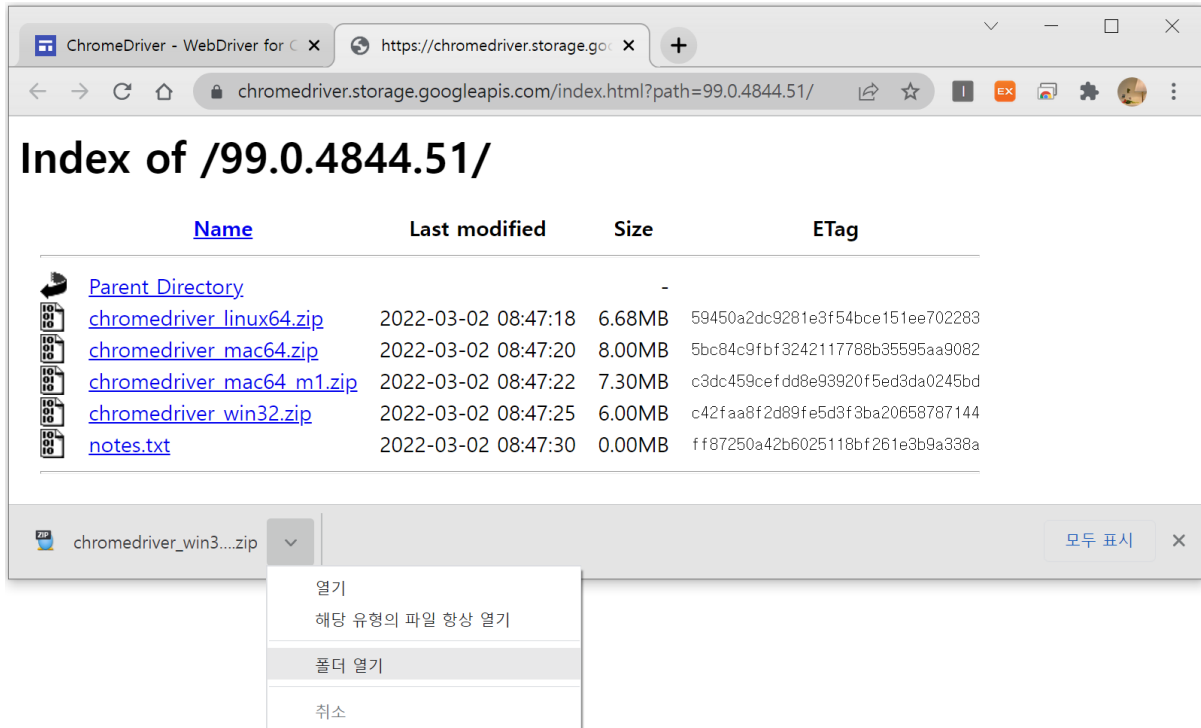
위 그림에서 크롬 드라이버 이름을 클릭하면 아래 그림과 같이 다양한 OS 용의 크롬 드라이버 목록이 나옵니다. 이 책에서는 가장 많이 사용되는 Windows 용으로 설명하겠습니다.

Step 2. 아래 그림에서 윈도우용인 **chromedriver_win32.zip** 을 클릭하여 다운로드 합니다.

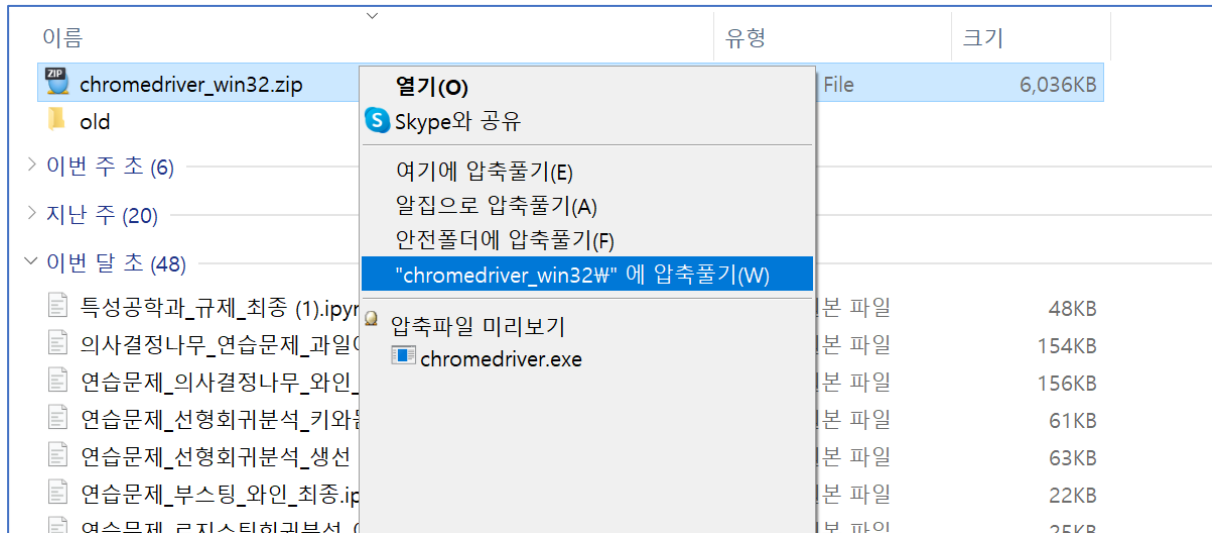
Index of /99.0.4844.51/

	Name	Last modified	Size	ETag
	Parent Directory		-	
	chromedriver_linux64.zip	2022-03-02 08:47:18	6.68MB	59450a2dc9281e3f54bce151ee702283
	chromedriver_mac64.zip	2022-03-02 08:47:20	8.00MB	5bc84c9fbf3242117788b35595aa9082
	chromedriver_mac64_m1.zip	2022-03-02 08:47:22	7.30MB	c3dc459cefdd8e93920f5ed3da0245bd
	chromedriver_win32.zip	2022-03-02 08:47:25	6.00MB	c42faa8f2d89fe5d3f3ba20658787144
	notes.txt	2022-03-02 08:47:30	0.00MB	ff87250a42b6025118bf261e3b9a338a

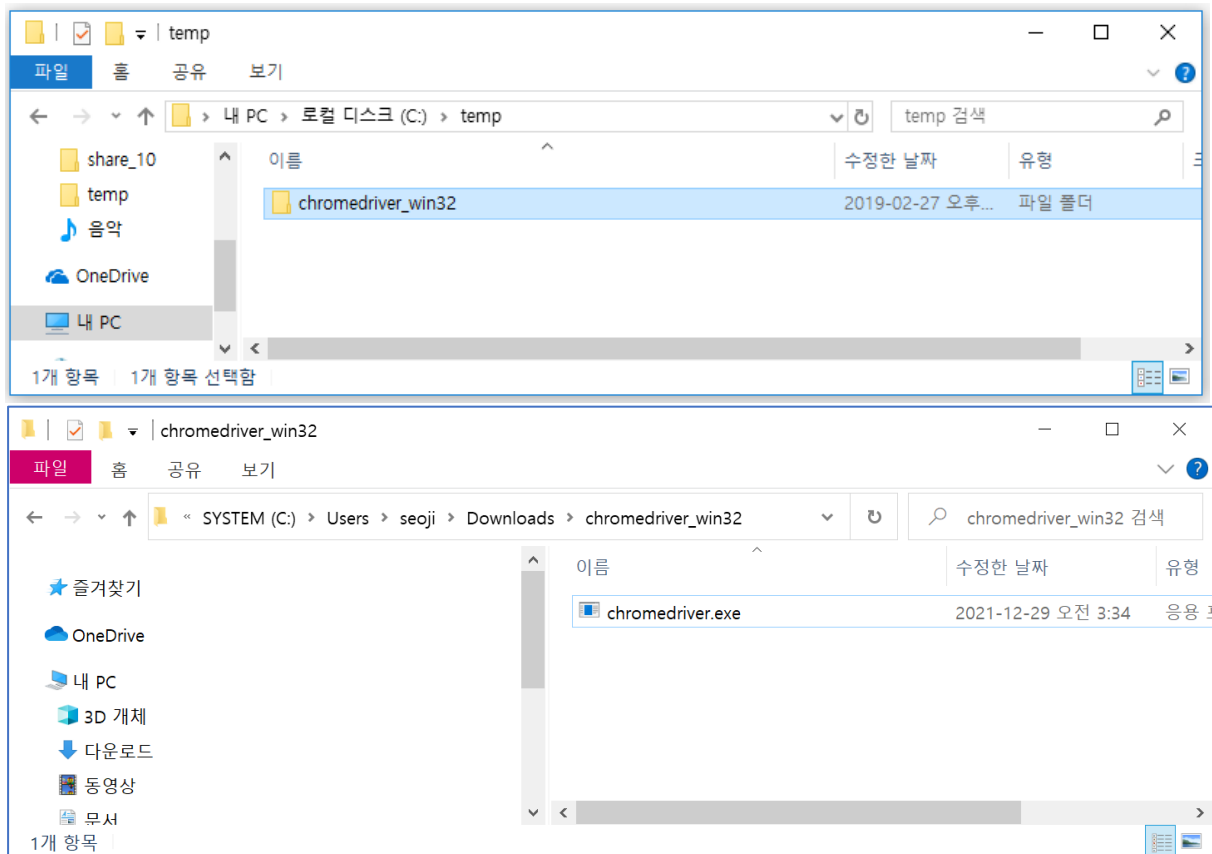
위 그림에서 보듯이 크롬 드라이버는 리눅스용과 macOS 용과 Windows 용이 있습니다. 이 책에서는 가장 많이 사용되는 Windows 용으로 다운로드를 받아서 사용하겠습니다. 위에서 chromedriver_win32.zip 링크를 눌러서 컴퓨터의 적당한 경로에 저장하세요.



다운로드가 완료되면 적당한 경로에 압축을 풀어 주세요.



압축이 풀리면 아래와 같이 chromedriver_win32 폴더가 생성되고 그 폴더 안에 chromedriver.exe 파일이 만들어 집니다.









위 상태에서 사용해도 되지만 실습의 편의를 위해 위 그림의 chromedriver.exe 파일을 복사한 후 c:\wpy_temp\w 폴더로 복사를 하겠습니다. 반드시 위 그림과 같이 경로를 설정할 필요는 없지만 이 책에서는 위 경로를 기준으로 설명을 진행하겠습니다.

만약 다른 경로에 chrome driver를 설정하셨다면 나중에 크롤링을 할 때 chromedriver.exe 파일을 사용하니까 경로 잘 기억해주세요 ^^






[파이썬 능력자 너도 될 수 있어~! - 서진수 저 -]

[macOS 사용자용 설정]

Index of /95.0.4638.54/

	Name	Last modified	Size	ETag
	Parent Directory		-	
	chromedriver_linux64.zip	2021-10-28 07:50:11	9.10MB	2dd873d43b446b88c174c35c9481d4be
	chromedriver_mac64.zip	2021-10-28 07:50:14	7.82MB	583ab0d55efc6827936f551bb4b49a21
	chromedriver_mac64_m1.zip	2021-10-28 07:50:16	7.39MB	a84ce737822297ed009895de34a9dd6
	chromedriver_win32.zip	2021-10-28 07:50:18	5.73MB	1190c1a1bb3638e89102e75ec8429ebe
	notes.txt	2021-10-28 07:50:23	0.00MB	63eb0eb24f7a7f6145ac6e9e47e92819

위 종류에서 mac64나 m1칩을 사용하는 분들은 mac64_m1zip 파일을 다운로드 받습니다.

	Name	Last modified	Size	ETag
	Parent Directory		-	
	chromedriver_linux64.zip	2021-11-08 09:38:00	9.56MB	25ae658c2b95f0c2008ba60380ac1bb4
	chromedriver_mac64.zip	2021-11-08 09:38:02	7.86MB	1564a6378810def81d3b99921da8a7ce
	chromedriver_mac64_m1.zip	2021-11-08 09:38:04	7.44MB	aae63a9d0fb7dd14d0ccd9121f03de2d
	chromedriver_win32.zip	2021-11-08 09:38:07	5.79MB	733f57989469f9bb6e27c896b79ed458

열기

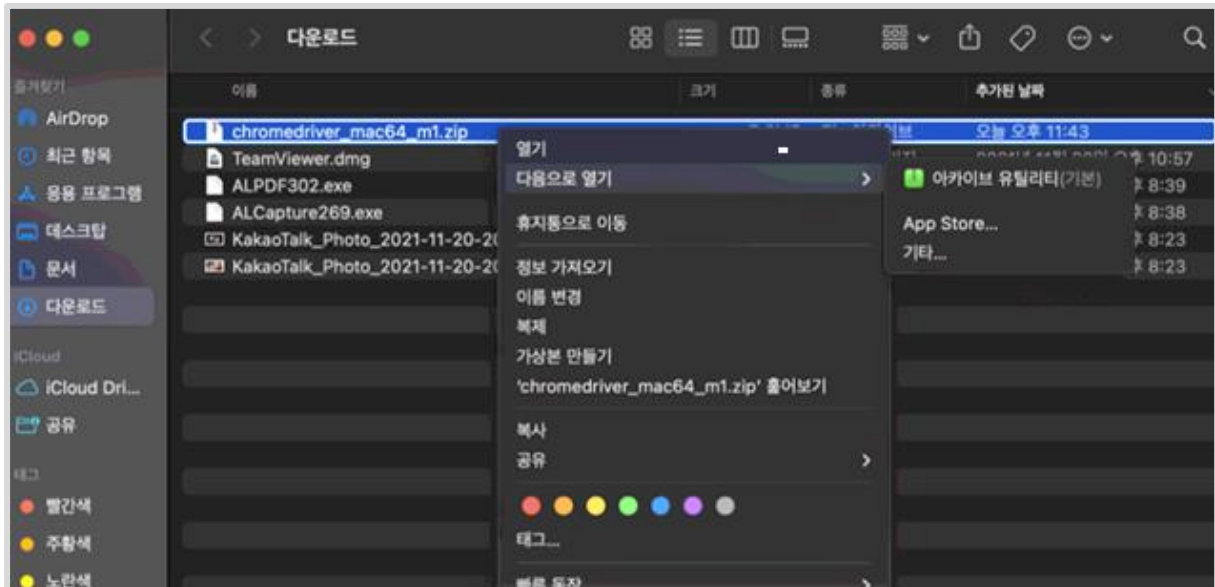
해당 유형의 파일 항상 열기

Finder에서 보기

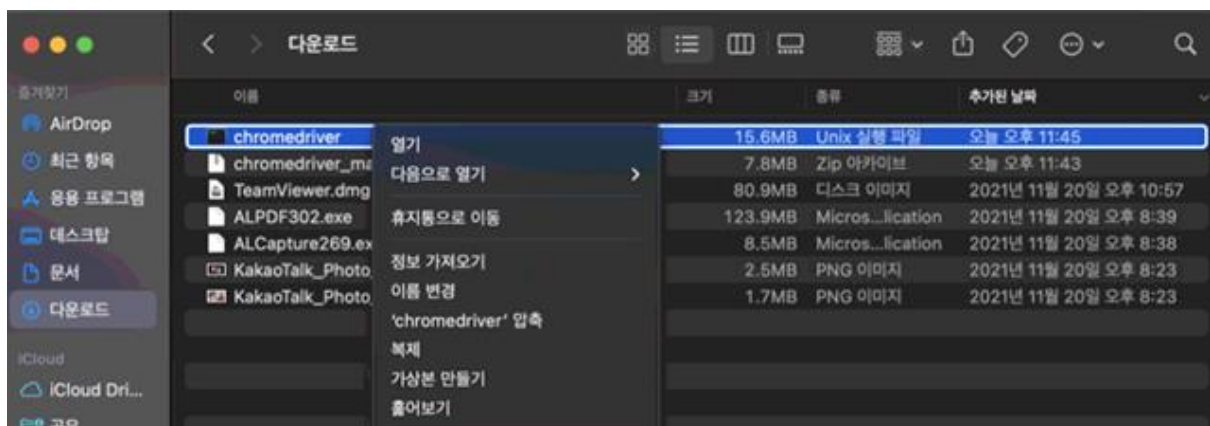
취소

chromedriver_ma....zip

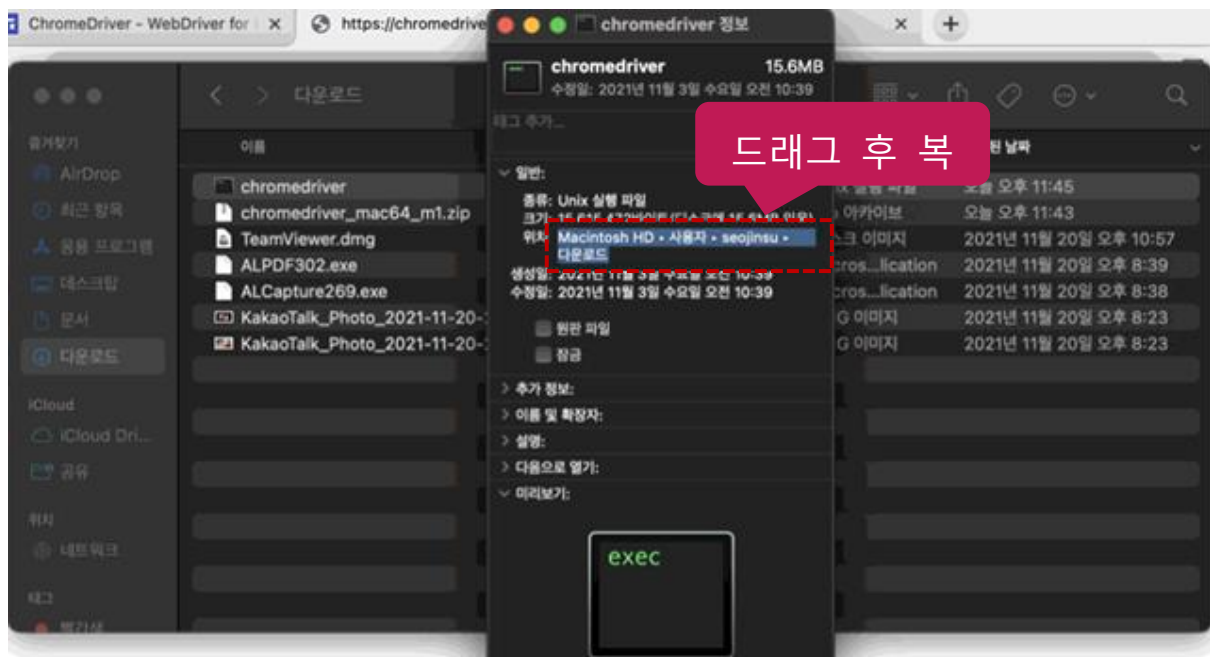
다운로드 받은 파일을 Finder에서 보기로 열어주세요.



위 그림과 같이 다운로드 받은 파일을 마우스 오른쪽 버튼으로 클릭한 후 다음으로 열기 -> 아카이브 유틸리티를 선택하여 압축을 풀어줍니다.



압축이 풀리면 chromedriver 파일이 생성되는데 이 파일을 마우스 오른쪽 버튼으로 클릭 후 정보 가져오기를 선택합니다.



정보 가져오기 창에서 위치 부분을 마우스로 드래그 한 후 아래 그림과 같이 소스코드의 경로 부분에 붙여넣기해서 사용하면 됩니다.

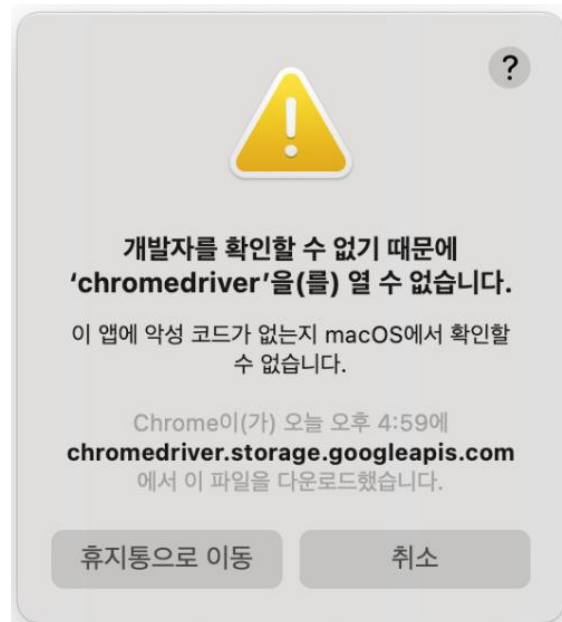
```

1  #Chap 13. 휴넷 사이트에서 특정 키워드로 강의 자료 검색하기
2
3  #Step 1. 필요한 모듈을 로딩합니다
4  from selenium import webdriver
5  import time
6
7  #Step 2. 사용자에게 검색 관련 정보들을 입력 받습니다.
8  print("=" * 100)
9  print(" 이 크롤러는 휴넷 사이트의 강의 자료 수집용 웹크롤러입니다.")
10 print("=" * 100)
11 query_txt = input('1. 수집할 자료의 키워드는 무엇입니까?(예: 파이썬): ')
12 print("\n")
13
14 #Step 3. 크롬 드라이버 설정 및 웹 페이지 열기
15 chrome_path = "/Users/seojinsu/Downloads/chromedriver"
16 driver = webdriver.Chrome(chrome_path)
17
18 url = 'https://www.hunet.co.kr/'
19 driver.get(url)
20 time.sleep(5)
21
22 #Step 4. 자동으로 검색어 입력 후 조회하기
23 driver.find_element_by_id("txtKeyword").click()
24 element = driver.find_element_by_id("txtKeyword")
25 element.send_keys(query_txt)
26 element.send_keys("\n")

```

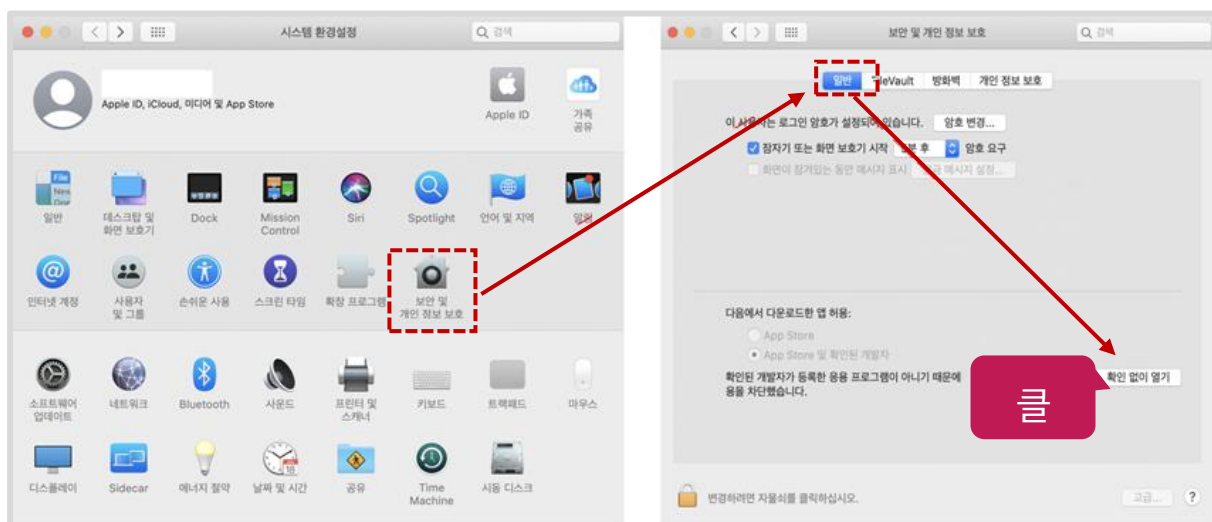
그리고 실행할 때 아래 그림과 같이 경고가 발생할 수 있습니다.

'chromedriver'은(는) Apple에서 악성 소프트웨어가 있는지 확인할 수 없기 때문에 열 수 없습니다.



이 경고는 macOS의 보안설정 때문에 발생하는 부분이라서 보안 설정을 수정해야 합니다.

아래 그림과 같이 launchpad의 보안 및 개인정보보호에 가서 일반 탭을 누르고 확인 없이 열기를 클릭해서 보안 설정을 수정해주세요



3. 검색창에 검색어를 입력 후 자동 검색하기 기능 구현하기

1) 작업 개요

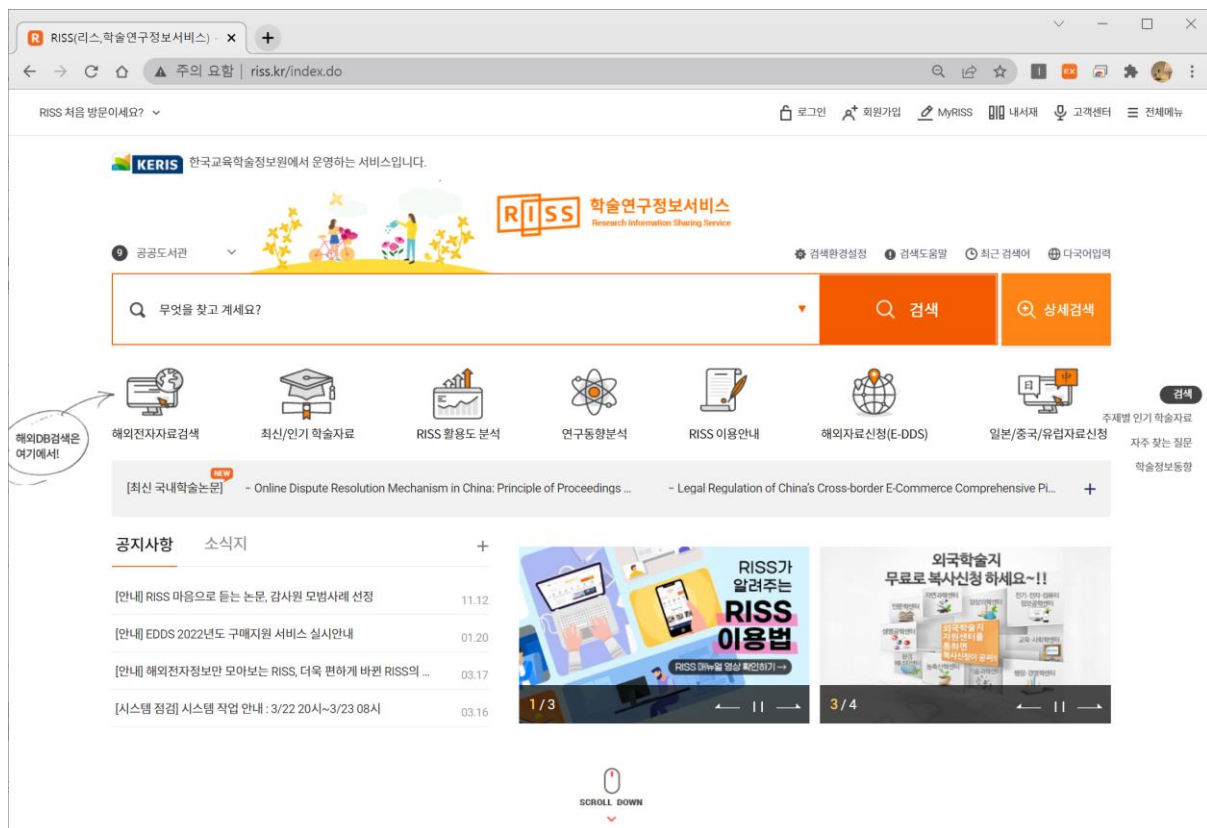
우리가 첫 번째 예제로 배울 내용은 검색창에 검색어를 자동으로 입력한 후 자동으로 검색하여 결과를 출력하게 만드는 것입니다.

즉 selenium 에게 **1.웹 페이지 열이라 -> 2.검색창을 찾아라 -> 3.검색어를 입력해라 -> 4.조회해라** 순서로 작업을 시키는 것이지요.

이 예제에서 반드시 배워야 하는 것은 웹 페이지에서 웹 브라우저의 개발자 도구를 사용해서 특정 메뉴나 항목(엘리먼트)의 이름이나 주소를 찾아내는 방법입니다.

이번 예제에서 사용할 웹 사이트는 한국교육학술정보원에서 운영하는 방대한 논문과 학술자료를 제공하는 대한민국 최고의 학술 정보사이트인 riss 입니다.

웹사이트 주소는 <http://www.riss.kr> 이고 아래 그림과 같습니다.



참고로 위 사이트의 모습은 수시로 바뀔 수 있습니다.

위 사이트에서 검색창에 특정 키워드를 입력한 후 검색을 실행하면 해당 키워드로 다양한 분야의 방대한 자료들이 조회됩니다.

우리는 selenium을 활용하여 자동으로 위 검색창에 “해양자원” 이라는 키워드를 입력 후 결과를 조회하도록 작업하겠습니다.

2) 일단 실행하기

앞의 작업 개요에서 살펴본 내용으로 작업을 하도록 파이썬으로 코드를 작성해야겠죠?
 이제부터 실제 파이썬 코드를 살펴보겠습니다.
 (아래의 소스 코드 코딩이 불편하신 분은 제가 제공해 드리는 소스코드를 사용하세요~)

#Chap 12. riss.kr 사이트에서 특정 키워드로 자동 검색하기

#Step 1. 필요한 모듈을 로딩합니다

```
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 time
```

#Step 2. 사용자에게 검색 관련 정보들을 입력 받습니다.

```
print("=" * 100)
print(" 이 크롤러는 riss 사이트의 논문 자료 수집용 웹크롤러입니다.")
print("=" * 100)
query_txt = input('1.수집할 자료의 키워드는 무엇입니까?(예: 해양자원): ')
print("\n")
```

#Step 3. 크롬 드라이버 설정 및 웹 페이지 열기

```
s = Service("c:/py_temp/chromedriver.exe")
driver = webdriver.Chrome(service=s)
```

```
url = 'https://www.riss.kr/'
driver.get(url)
time.sleep(5)
```

#Step 4. 자동으로 검색어 입력 후 조회하기

```
element = driver.find_element(By.ID,'query')
driver.find_element(By.ID,'query').click( )
element.send_keys(query_txt)
element.send_keys("\n")
```

위 소스 코드를 Shift + Enter 키를 눌러 실행하면 아래 그림과 같이 크롤링 할 검색어를 입력하는 창이 나오죠?

1. 수집할 자료의 키워드는 무엇입니까?(예: 해양자원):

=====

이 크롤러는 riss 사이트의 논문 자료 수집용 웹크롤러입니다.

=====

키워드를 입력하는 곳에 "해양자원"을 입력한 후 엔터키를 치면 Chrome 창이 새로 열리면서 위 코드의 18번 행에서 지정한 RISS 웹 페이지가 열리고 검색창에 "해양자원" 을 자동으로 입력한 후 검색하여 결과가 화면에 출력되는 것이 보이죠?

3) 소스 코드 설명

앞에서 실행한 코드를 자세하게 설명 드리겠습니다.

1 #Step 1. 필요한 모듈을 로딩 합니다

```
2 from selenium import webdriver
3 from selenium.webdriver.common.by import By
4 from selenium.webdriver.common.keys import Keys
5 from selenium.webdriver.chrome.service import Service
6 import time
```

위 코드의 4번행에서 selenium 모듈을 import 했습니다.

그리고 아래의 5-7행까지는 웹페이지에서 특정 엘리먼트를 찾거나 글자를 입력할 때 필요한 모듈을 불러오는 부분입니다

.import 명령에 대한 자세한 설명은 이 책의 모듈 부분에 있으니까 참고하세요~

그 후 Step 2의 11번 행에서 사용자에게 검색어를 입력 받아 query_txt 라는 변수에 저장합니다.

7 #Step 2. 사용자에게 검색 관련 정보들을 입력 받습니다.

```
8 print("=" *100)
9 print(" 이 크롤러는 riss 사이트의 논문 자료 수집용 웹크롤러입니다.")
10 print("=" *100)
11 query_txt = input('1.수집할 자료의 키워드는 무엇입니까?(예: 해양자원): ')
12 print("\n")
```

이렇게 작성 후 Shift+Enter 키를 입력하면 아래와 같이 검색어를 입력 받는 창이 나오는데 "해양자원" 을 입력 한 후 Enter 키를 치세요.

1. 수집할 자료의 키워드는 무엇입니까?(예: 해양자원):

=====

이 크롤러는 riss 사이트의 논문 자료 수집용 웹크롤러입니다.

=====

사용자가 입력한 "해양자원" 키워드가 11번 행의 query_txt 변수에 저장되어 있겠죠?

이제 아래 그림처럼 크롬 드라이버를 설정하고 웹 페이지에 접속합니다.

13 #Step 3. 크롬 드라이버 설정 및 웹 페이지 열기

```
14 s = Service("c:/py_temp/chromedriver.exe")
15 driver = webdriver.Chrome(service=s)
16
17 url = 'https://www.riss.kr/'
18 driver.get(url)
19 time.sleep(5)
20 driver.maximize_window( )
```

위 그림에서 14번 행의 Service()에는 chromedriver.exe 파일이 있는 경로를 적어주면 됩니다. 이 책에서는 위 경로이지만 혹시 다른 경로에 chromedriver.exe 파일이 있는 분들은 그 경로로 변경해서 사용하시면 됩니다.

그리고 15번 행에서 webdriver.Chrome() 함수로 웹 드라이버는 크롬 드라이버를 사용한다고 지정하면 됩니다. 만약 다른 웹 드라이버를 사용할 경우 이 부분에서 다른 웹 드라이버를 지정하면 됩니다.

17번 행에서 우리가 접속할 홈페이지의 주소를 지정한 후 18번 행에 있는 driver.get("URL 주소") 명령으로 주어진 URL 주소의 웹 페이지를 OPEN 합니다. 사람이 웹 페이지를 열기 위해 웹 브라우저 실행 후 주소창에 주소를 입력하는 것과 똑같은 효과를 내는 명령어입니다.

여러분들이 크롤링하거나 열기를 원하는 주소로 바꾸고 실행해 보는 것도 재미있겠죠?

19번 행의 time.sleep() 명령은 인터넷 속도에 따라 웹 페이지가 열리는 속도가 다르기 때문에 driver.get() 명령으로 웹 페이지를 열도록 시킨 후 다 열릴 때까지 기다리는 명령인데 인터넷 속도가 느릴 경우 이 명령을 사용하지 않으면 에러가 자주 발생할 정도로 아주 중요한 역할을 합니다. 괄호안에 기다릴 시간을 초 단위로 적으면 됩니다.

그리고 20번 행은 chrome driver 로 열린 창을 전체 창 크기로 확대하라는 의미입니다.

웹 페이지가 열렸다면 이제 키워드를 입력한 후 검색을 해야겠죠?

아래 코드를 볼까요?

21 #Step 4. 자동으로 검색어 입력 후 조회하기

```
22 element = driver.find_element(By.ID,'query')
23 driver.find_element(By.ID,'query').click( )
24 element.send_keys(query_txt)
25 element.send_keys("\n")
```

위 코드에서 22번 행이 검색어를 입력하는 검색 창 정보를 지정하는 명령입니다.
그리고 23번 행에서 검색 창을 클릭한 후 24번 행에서 검색어를 입력하고 25번 행에서 엔터키를 입력하여 검색을 진행하게 됩니다.

4) 개발자 도구 사용하기

우리는 앞의 실습에서 아래와 같이 입력하고 Shift+Enter를 실행하면 자동으로 크롬이 실행되면서 RISS의 웹 페이지가 열린 후 검색창에 검색어가 입력되는 것을 확인했습니다.

21 #Step 4. 자동으로 검색어 입력 후 조회하기

```
22 element = driver.find_element(By.ID,'query')
23 driver.find_element(By.ID,'query').click( )
24 element.send_keys(query_txt)
25 element.send_keys("\n")
```

위 코드에서 23번 행에 보면 `driver.find_element(By.ID,'query').click()` 명령이 보이죠?
이 줄의 의미는 selenium 에게 현재 보이는 웹페이지에서 ID값이 "query" 라는 것을 찾아서 클릭하라는 뜻입니다. 여기서 ID값이 "query" 가 과연 무엇이고 어떻게 찾을까요?

앞에서 언급한 작업 순서 중에 현재 페이지에서 검색창에 검색어를 입력한다고 했는데 바로 현재 페이지의 HTML 코드에서 검색창의 ID 값이 query입니다. 이처럼 우리가 웹페이지에서 클릭하거나 값을 입력하는 등의 작업을 하는 대상을 **엘리먼트(element)**라고 합니다.

그럼 검색창의 ID 가 query 라는 것은 어떻게 찾았을까요?

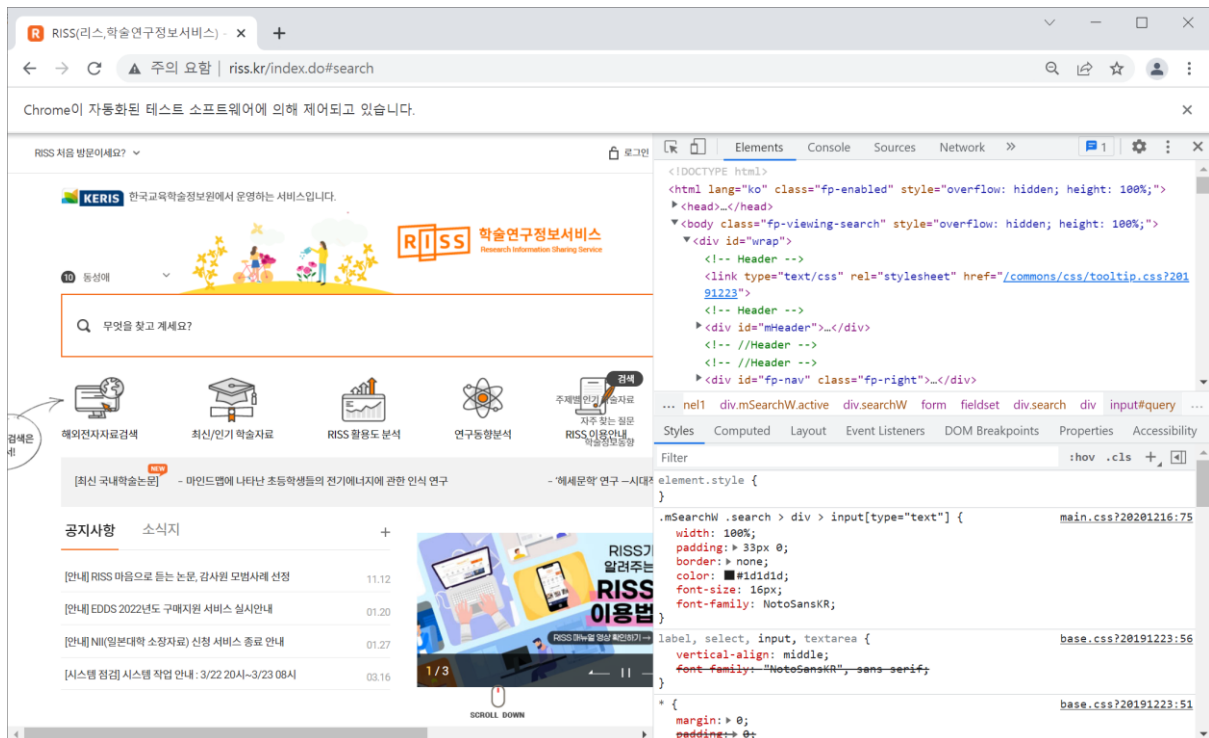
바로 웹 브라우저에서 제공하는 기능인 개발자 도구를 사용해서 쉽게 찾을 수 있습니다.

지금 바로 실습으로 크롬에서 개발자 도구를 이용해서 특정 엘리먼트의 이름이나 ID 등을 찾는 방법을 소개해 드리겠습니다.

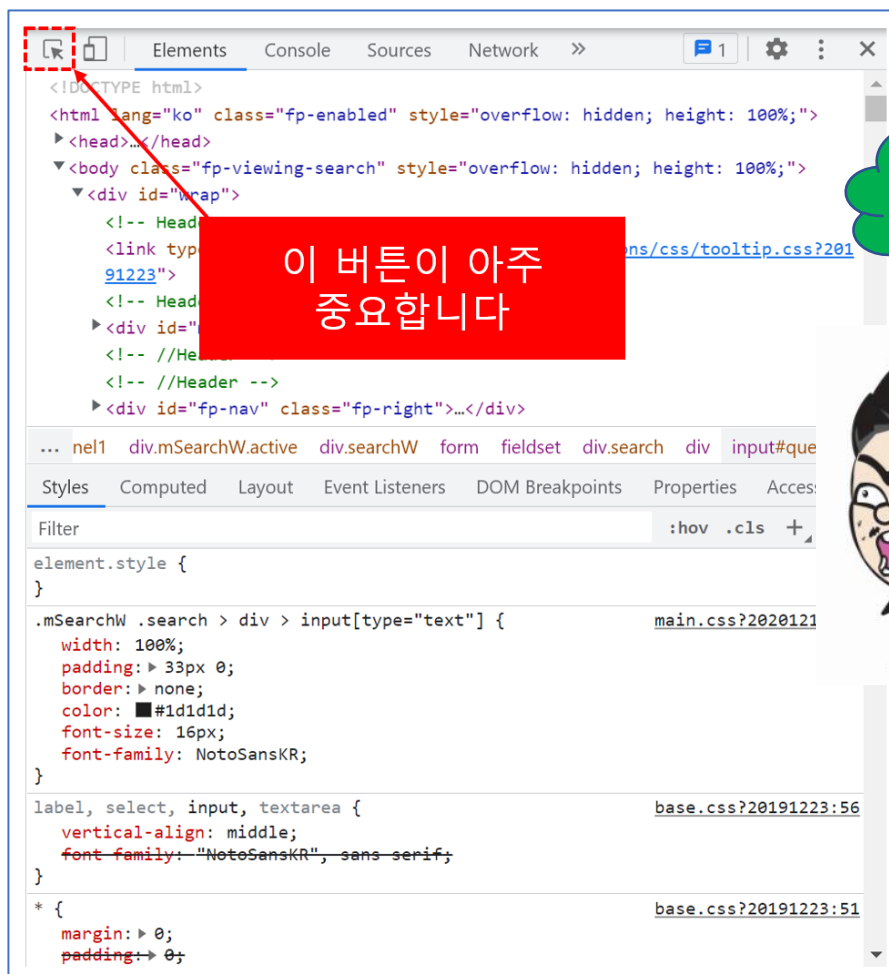
먼저 현재 RISS 웹 페이지가 있는 화면에서 F12 번 키나 Ctrl+Shift+I 키를 누르세요.

그러면 아래와 같이 화면이 좌/우로 분할이 되면서 보일 거예요.

(크롬 드라이버일 경우 좌/우로 분할되고 인터넷 익스플로러일 경우 상/하로 분할됩니다)



화면이 너무 작아서 잘 안보일 수도 있는데 위 화면에서 오른쪽 화면만 확대해서 다시 보겠습니다.



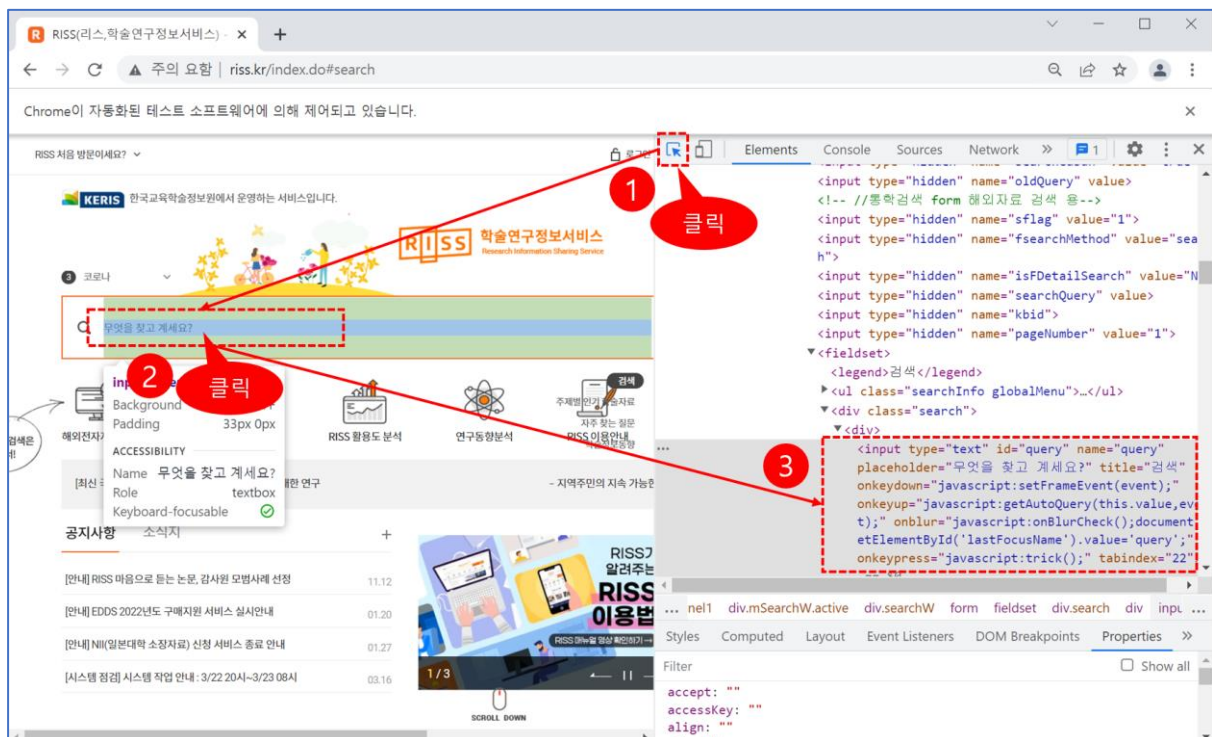
[파이썬 능력자 너도 될 수 있어~! - 서진수 저 -]

위 화면이 현재 웹 페이지의 HTML 전체 코드를 보여주는 창입니다.

이 창에서 우리가 원하는 엘리먼트 정보를 많이 찾게 될 거예요.

위 창에서 특정 엘리먼트를 선택하는 버튼을 누른 후 오른쪽 웹 페이지에서 해당 엘리먼트를 선택하면 오른쪽 창에서 해당 엘리먼트의 HTML 코드를 보여줍니다.

말로 하면 어려우니까 아래 그림의 순서를 잘 보세요.



위 그림에서 HTML소스코드가 있는 3번 부분을 보면 id="query" 부분이 보이죠?

이렇게 개발자 도구를 활용해서 특정 버튼이나 메뉴의 name 값이나 id 값을 찾으면 됩니다.

여기서는 해당 엘리먼트의 id 나 name 값이 있어서 둘 중에서 id 값을 사용했는데 만약 id 값이 나 name 값이 없을 경우도 있어서 아래의 다양한 방법들을 사용하여 특정 엘리먼트를 클릭하거나 값을 입력할 수도 있습니다.

우선 id 값을 이용하는 방법부터 살펴보고 나머지 방법들은 다른 실습에서 사용하겠습니다.

참고하세요 - 웹 페이지의 특정 element 정보를 지정할 때 많이 사용하는 방법들

- find_element(By.ID, 'id값')
- find_element(By.NAME, 'name값')
- find_element(By.XPATH, 'xpath값')
- find_element(By.LINK_TEXT, 'text값')
- find_element(By.TAG_NAME, 'tag name값')

이번 챕터의 내용들이 앞으로 웹 크롤러를 만들 때 아주 중요하게 사용될 내용들이니까 막히는 부분이 없도록 열심히 연습해 주세요~

[파이썬 능력자 너도 될 수 있어~! - 서진수 저 -]

4. 연습 문제로 실력 굳히기

1. 네이버 사이트의 검색창에 “서진수 빅데이터” 키워드를 입력한 후 검색을 실행하도록 코드를 작성하세요
2. 다음 사이트의 검색창에 “서진수 빅데이터” 키워드를 입력한 후 검색을 실행하도록 코드를 작성하세요.
3. 구글 사이트의 검색창에 “서진수 빅데이터” 키워드를 입력한 후 검색을 실행하도록 코드를 작성하세요
4. 대한민국 구석구석 사이트(<https://korean.visitkorea.or.kr>) 에 자동 접속하여 검색창에 “제주도” 키워드를 입력하여 자동 검색하도록 코드를 작성하세요.
5. 대한항공 홈페이지 (<https://www.koreanair.com/>) 에 자동 접속하여 오른쪽 상단의 돋보기 버튼을 클릭하여 “제주도” 키워드를 입력하여 자동 검색하도록 코드를 작성하세요.
6. 가천대학교 홈페이지(<https://www.gachon.ac.kr/>) 에 자동 접속하여 오른쪽 상단의 돋보기 버튼을 클릭하여 “장학금” 키워드를 입력하여 자동 검색하도록 코드를 작성하세요.
7. 서울대학교 홈페이지(<https://www.snu.ac.kr/>) 에 자동 접속하여 왼쪽 아래부분의 돋보기 버튼을 클릭하여 “장학금” 키워드를 입력한 후 자동 검색하도록 코드를 작성하세요

이번 장에서 배운 내용들을 완전히 습득하셔서 꼭 여러분들의 실력으로 만드세요~



데이터쟁이 서진수가 여러분을 응원합니다~!

[파이썬 능력자 너도 될 수 있어~! - 서진수 저 -]