

## [Python] python 에서 Seleium을 통한 동적 크롤링 - 3 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-27 오전 8:32

URL: <https://continuous-development.tistory.com/114?category=736681>

Python

# [Python] python 에서 Seleium을 통한 동적 크롤링 - 3

2020. 10. 8. 17:04 수정 삭제 공개

동적으로 움직이면서 크롤링하기

NAVER 자동차

자동차 홈 | 자동차 | **모터사이클**

모터사이클 모델 비교

최신모델 인기모델

모터사이클 전체 제조사 >

- HONDA 혼다
- BMW BMW
- Harley-Davidson 할리데이비슨
- Yamaha 야마하
- Ducati 두카티
- Kawasaki 카와사키

스쿠터 >

네이키드 >

크루저 >

2020 BMW M1000RR

가격정보없음




출력 212.0ps / 14500rpm

[가격/제원 자세히보기](#)

시판 모델 단종모델 레이싱/컨셉 클래식바이크


✓ 인기순 최신출시순 출력높은순 가격낮은순

출시년도: 기간 전체

 <b>2020 BMW C400GT</b> 출시 1,110만원 출력 34.0ps / 7500rpm 연료 가솔린	 <b>2020 Honda 슈퍼브</b> 출시 240만원 출력 9.1ps / 7500rpm 연료 가솔린	 <b>2020 Honda PCX125</b> 출시 403만원 출력 12.2ps / 8500rpm 연료 가솔린
--	--	--

추천 모터사이클

조화로운 출력높은순 최신출시순

1  2020 BMW C400GT

네이버에 해당 사이트를 크롤링해보자.

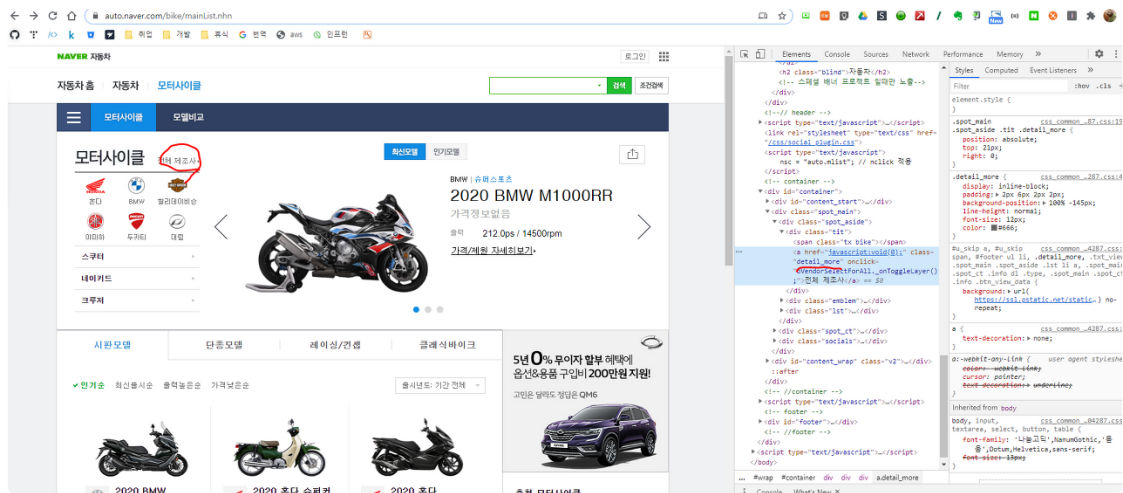
```
In [87]: from selenium import webdriver
import time
path = './chromedriver.exe'
driver = webdriver.Chrome(path)
driver.get('https://auto.naver.com/bike/mainList.nhn')
```

- 전체 제조사 엘리먼트를 찾아서 마우스 클릭하고 그 결과물을 반환 받는다.
- 바이크 제조사 정보를 크롤링
- 다음 페이지 버튼이 활성화 되어 있으면 클릭하고 추가적으로 제조사 정보를 반환받는다.
- 다음 페이지 버튼이 비활성화 되어 있으면 크롤링을 중단한다.

해당 사이트에서 전체 제조사라는 것을 클릭하고 그 나온 결과를 크롤링 하려 한다.

사이트는 <https://auto.naver.com/bike/mainList.nhn> 다.

저 전체 제조사라는 버튼을 클릭하려 한다.



```
In [88]: # 바이크 제조사 전체 페이지 클릭 구현
```

```
In [101]: bikeElementBtn = driver.find_element_by_css_selector('.detail_more')
# bikeElementBtn = driver.find_element_by_css_selector('#container > div, con')
bikeElementBtn.click()
```

해당 엘리먼트를 가져오는데 class 부분을 가져온다. 이렇게 까지 하면 해당 부분을 클릭하게 된다.

모터사이클

모델비교

1/2

<

>

BMW

KR모터스

KTM

MV아구스타

SYM

가와사키

gogoro 고고로

니우

다임러

닷지

대림

두카티

로얄엔필드

모토구찌

베넬리

베스피

베타

벡트릭스

브라모

블루사크

비모터

빅토리

스즈키

아치

아프탈리아

아마하

우랄

italjet 이탈리아

인디언

재로

질레라

켄암 BRP

킴코

트라이엠프

폴라리스

장르

네이키드

멀티퍼프스

비즈니스

슈퍼모터드

슈퍼스포츠

스쿠터

스포츠

엔듀로

전기모터

크루저

투어러

트라이크

ATV

그러면 결과가 이렇게 되고 다음에는 해당 부분의 제조사를 모두 가져오기 위해 li 부분을 가져온다.

```
In [99]: time.sleep(3)

In [115]: bikeElements = driver.find_elements_by_css_selector('#_vendor_select_layer > div > div.maker_group > div.emblem_area > ul > li')
# bikeElements = driver.find_elements_by_css_selector('#_vendor_select_layer>div>div.maker_group>div.emblem_area>ul>li')
bikeElements

Out [115]: [<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="808595dc-df5e-47b3-8127-558b2ecab156")>,
<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="1365fbca-5796-4065-84c1-f8c34fee9558")>,
<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="f10a3666-177f-472a-af7d-35585a6a037f")>,
<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="156d2511-3ab1-4e6f-a309-787082f6c323")>,
<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="41458531-c0c4-40ba-8485-d361387cd1cb")>,
<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="a82b9c28-8aa4-4c0c-a32f-668bb319ec7b")>,
<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="c565e344-5e80-4d9e-b1b0-99089f1d548a")>,
<selenium.webdriver.remote.webelement.WebElement (session="3a4f9104cb8e6e0a958b9e0cab8c57d1", element="a4d28473-37f8-4545-a182-cl0d0368b5c2
```

해당 부분의 li는 중복되는 경우가 있어서 아래와 같이 경로를 세부적으로 설정하여 만들어 주었다.

```
In [117]: for bike_li in bikeElements:
companyName = bike_li.find_element_by_tag_name('span').text
print(companyName)

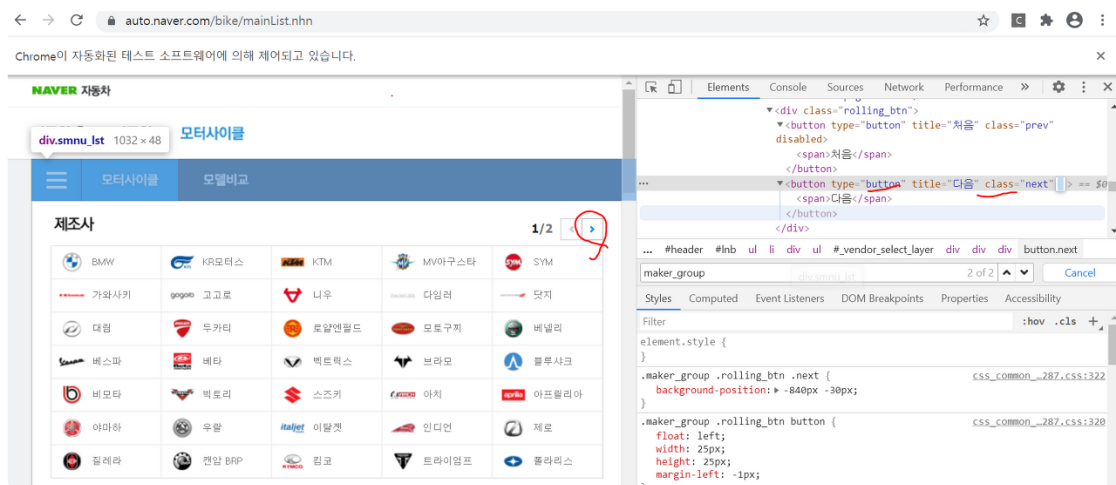
BMW
KR모터스
KTM
MV아구스타
SYM
```

그다음은 가져왔던 bikeElements를 리스트 형태로 되어 있는데 이것을 for문으로 하나씩 뽑고 그중에 내가 필요한 span을 가져온다.

```
In [131]: for bike_li in bikeElements:
            companyName = bike_li.find_element_by_tag_name('span').text
            print(companyName.strip())
            if companyName == '':
                break
            # 이미지
            img = bike_li.find_element_by_tag_name('img').get_attribute('src')
            print(img)
            link = bike_li.find_element_by_tag_name('a').get_attribute('href')
            print(link)
            time.sleep(3)
```

```
제호
https://imgauto-phinf.pstatic.net/20200211_86/auto_1581399544042tVyrG_PNG/20200211143900_Aak3gllX.png?type=f30_30
https://auto.naver.com/bike/mnfcoMain.nhn?mfcoNo=45
질레라
https://imgauto-phinf.pstatic.net/20170707_150/auto_14994035347220bKh_PNG/20170707135852_N2uEr3sk.png?type=f30_30
https://auto.naver.com/bike/mnfcoMain.nhn?mfcoNo=6
캠알 BFP
https://imgauto-phinf.pstatic.net/20170707_228/auto_1499403938849PlyMC_PNG/20170707140536_X569kcW8.png?type=f30_30
https://auto.naver.com/bike/mnfcoMain.nhn?mfcoNo=35
킴코
https://imgauto-phinf.pstatic.net/20200211_105/auto_1581393647952v08ut_PNG/20200211130046_kyHvtGAm.png?type=f30_30
https://auto.naver.com/bike/mnfcoMain.nhn?mfcoNo=11
트라이엠프
https://imgauto-phinf.pstatic.net/20200306_108/auto_158347721991817tk_PNG/20200306154654_YfKRCDoV.png?type=f30_30
https://auto.naver.com/bike/mnfcoMain.nhn?mfcoNo=31
폴라리스
https://imgauto-phinf.pstatic.net/20200306_92/auto_1583477538592tLog2_PNG/20200306155212_sPrEH7Nu.png?type=f30_30
https://auto.naver.com/bike/mnfcoMain.nhn?mfcoNo=27
```

추가적으로 이미지와 해당 링크에 대한 url을 가져오자



이제 다음은 저 화살표 버튼을 클릭해서 다음 장에 있는 제조사를 가져오는 것을 해보자. 그전에 현재 위치에서 다음 페이지가 있는지를 확인하고 만약 있다면 버튼을 누르게 하고 아닐 경우에는 작동이 안 되게 만들어야 된다.

```
In [145]: nextBtn = driver.find_element_by_css_selector('#_vendor_select_layer > div > div.maker_group > div.rolling_btn > button.next')
            isFlag = nextBtn.is_enabled()
            print(isFlag)

True
```

nextBtn이라는 것을 만들고 selector를 통해 해당 경로에 있는 button next를 가져온다. 해당 클래스에서, 을붙이고 title을 넣어준다.

그리고 이게 다음 장이 있는지의 여부에 따라 넘어가야 하기 때문에 is\_enabled()를 통해 있는지를 확인한다.

이때 True를 반환하면 넘어갈 페이지가 있다는 뜻이다.

```
In [146]: if isFlag :
            print('next page exist~~~')
            nextBtn.click()
            bikeElements = driver.find_elements_by_css_selector('#_vendor_select_layer > div > div.maker_group > div.emblem_area > ul > li')

            for bike_li in bikeElements:
                companyName = bike_li.find_element_by_tag_name('span').text
                print(companyName.strip())
                if companyName == '':
                    break
            # 이미지
            img = bike_li.find_element_by_tag_name('img').get_attribute('src')
            print(img)
            link = bike_li.find_element_by_tag_name('a').get_attribute('href')
            print(link)

next page exist~~~
```

마지막으로 if 조건문으로 다음 값이 있을 때 사용하고 위에서 돌렸던 소스들을 가져온다.

## 'Python' 카테고리의 다른 글

[Python] Numpy에 대한 기초 정리와 사용법 정리

[Python] python 에서 Seleium을 통한 동적 크롤링 - 4

**[Python] python 에서 Seleium을 통한 동적 크롤링 - 3**

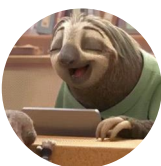
[Python] python 에서 Seleium을 통한 동적 크롤링 - 2

[Python] python 에서 Seleium을 통한 동적 크롤링 - 1

[Python] python 에서 Seleium을 설치 방법

python 동적 크롤링

파이썬 동적 크롤링



나무늘보스

혼자 끄적끄적하는 블로그 입니다.

