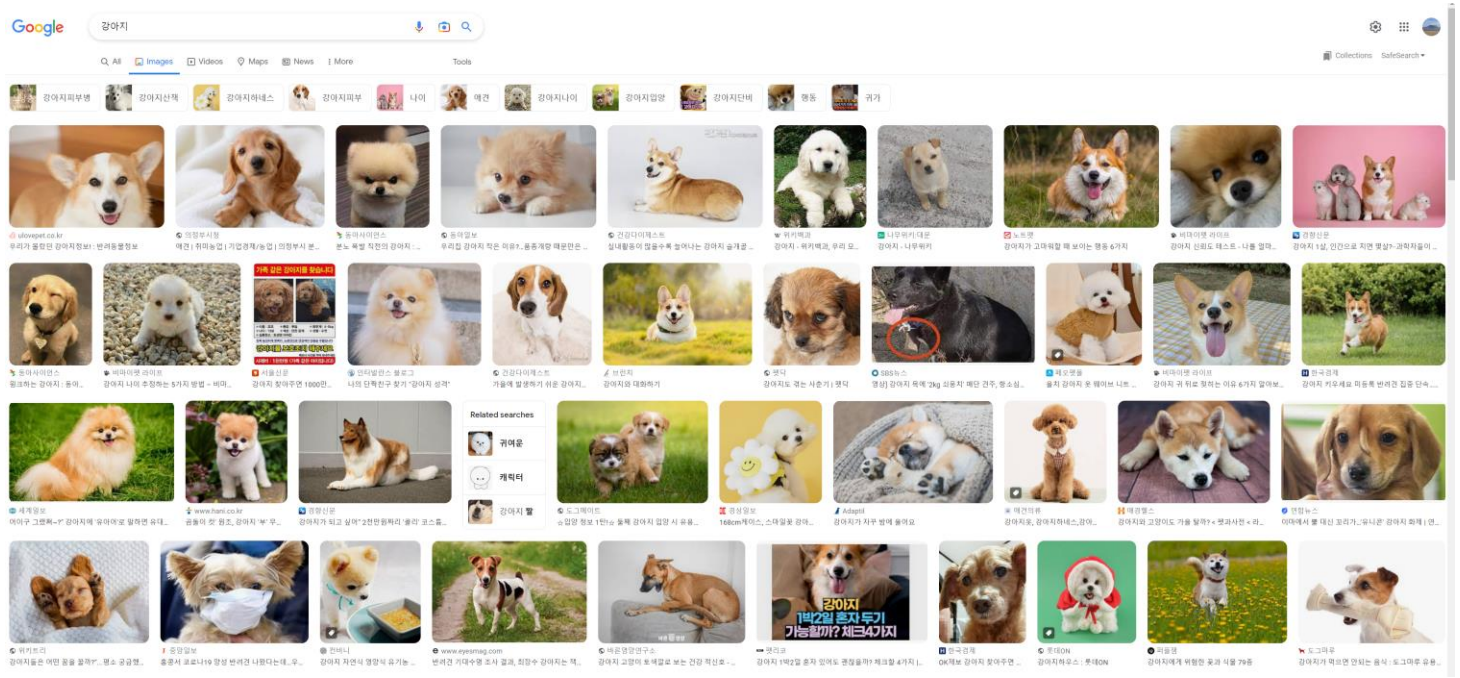
# Web crawling with Selenium

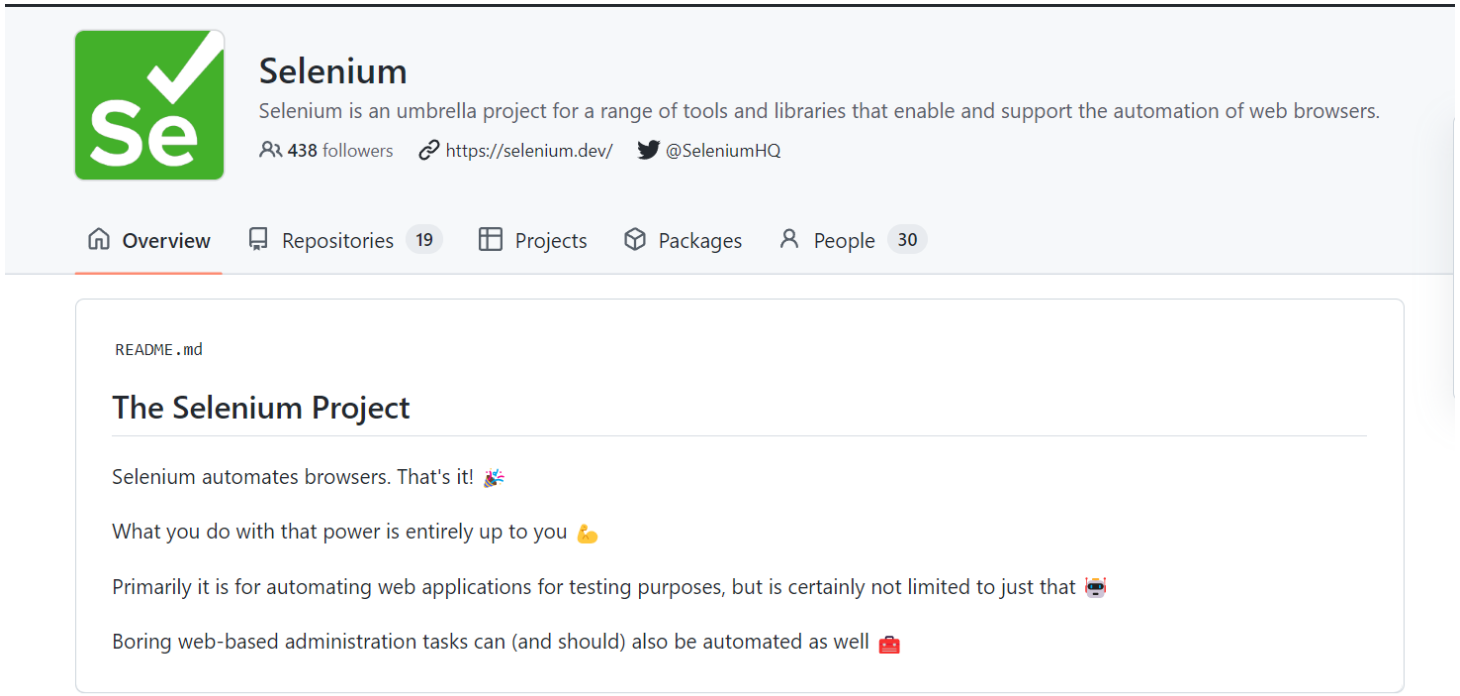
# Dynamic Rendering of Web pages



# Authentification

# Selenium

- Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation.



# Selenium Install

- Python bindings for Selenium

## selenium 3.4.3

*Python bindings for Selenium*

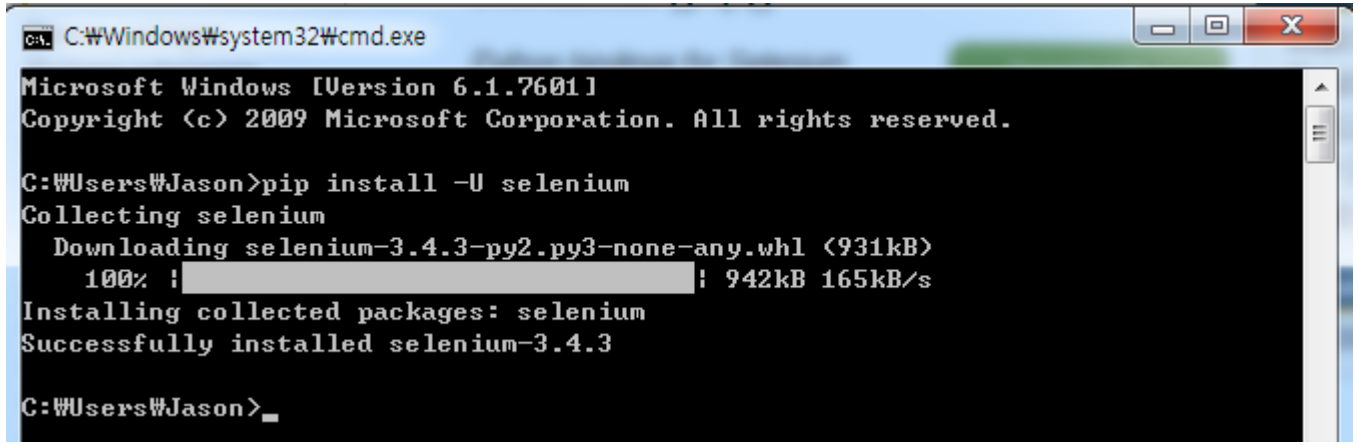
Downloads ↓

### Introduction

Python language bindings for Selenium WebDriver.

The *selenium* package is used to automate web browser interaction from Python.

**Home**: http://www.seleniumhq.org
**Docs**: selenium package API
**Dev**: https://github.com/SeleniumHQ/Selenium
**PyPI**: https://pypi.python.org/pypi/selenium
**IRC**: **#selenium** channel on freenode

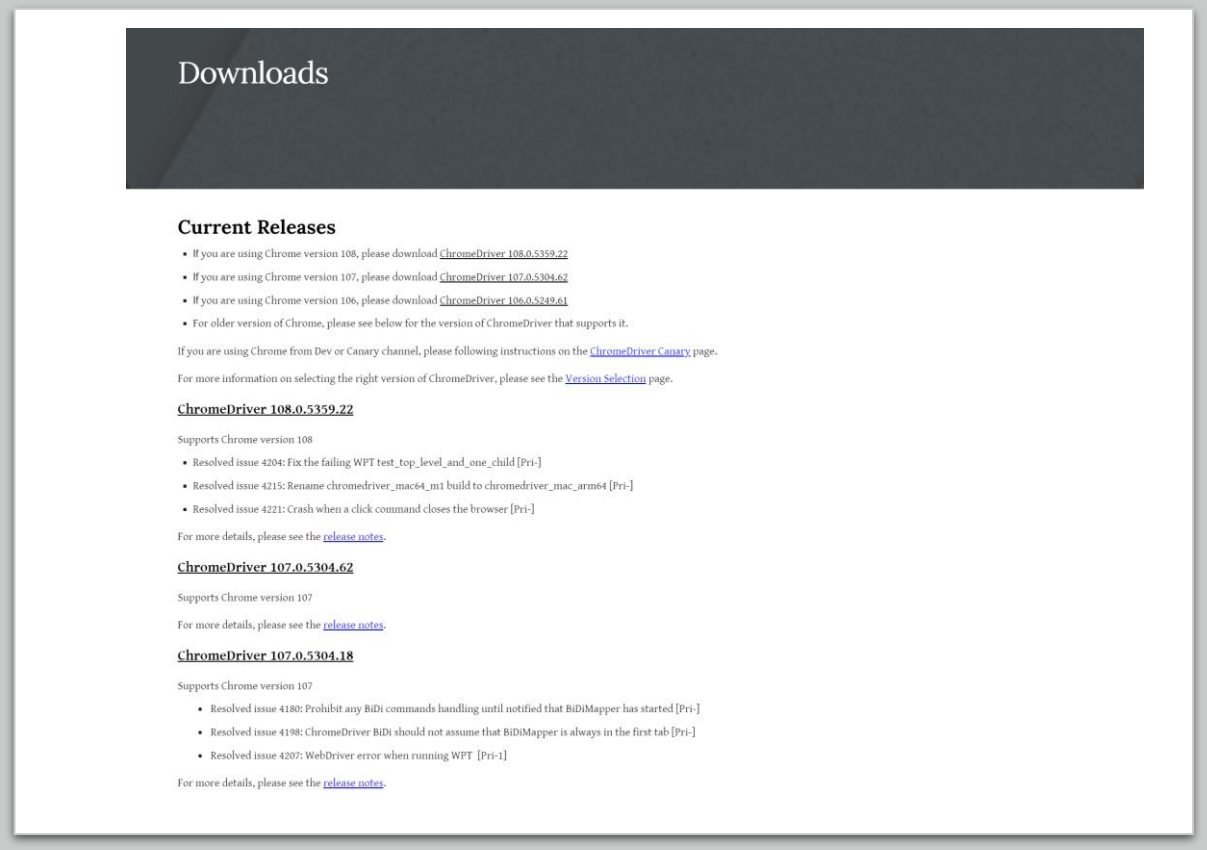



윈도우 명령 프롬프트를 실행한 후 다음 명령어를 입력

pip install –U selenium

# ChromeDriver - WebDriver for Chrome

- https://chromedriver.chromium.org/downloads



Downloads

**Current Releases**

- If you are using Chrome version 108, please download ChromeDriver 108.0.5359.22
- If you are using Chrome version 107, please download ChromeDriver 107.0.5304.62
- If you are using Chrome version 106, please download ChromeDriver 106.0.5249.61
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the ChromeDriver Canary page.

For more information on selecting the right version of ChromeDriver, please see the Version Selection page.

**ChromeDriver 108.0.5359.22**

Supports Chrome version 108

- Resolved issue 4204: Fix the failing WPT test_top_level_and_one_child [Pri-]
- Resolved issue 4215: Rename chromedriver_mac64_m1 build to chromedriver_mac_arm64 [Pri-]
- Resolved issue 4221: Crash when a click command closes the browser [Pri-]

For more details, please see the release notes.

**ChromeDriver 107.0.5304.62**

Supports Chrome version 107

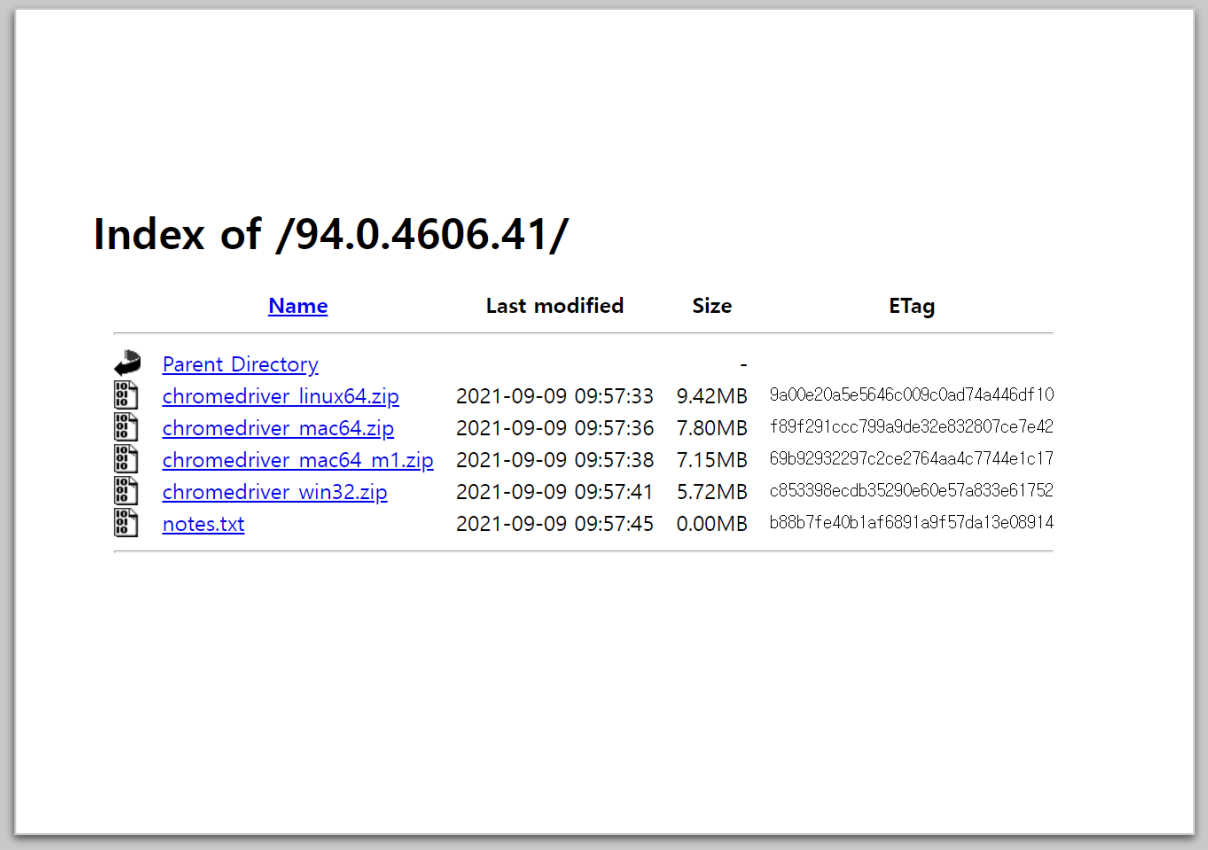
For more details, please see the release notes.

**ChromeDriver 107.0.5304.18**

Supports Chrome version 107

- Resolved issue 4180: Prohibit any BiDi commands handling until notified that BiDiMapper has started [Pri-]
- Resolved issue 4198: ChromeDriver BiDi should not assume that BiDiMapper is always in the first tab [Pri-]
- Resolved issue 4207: WebDriver error when running WPT [Pri-1]

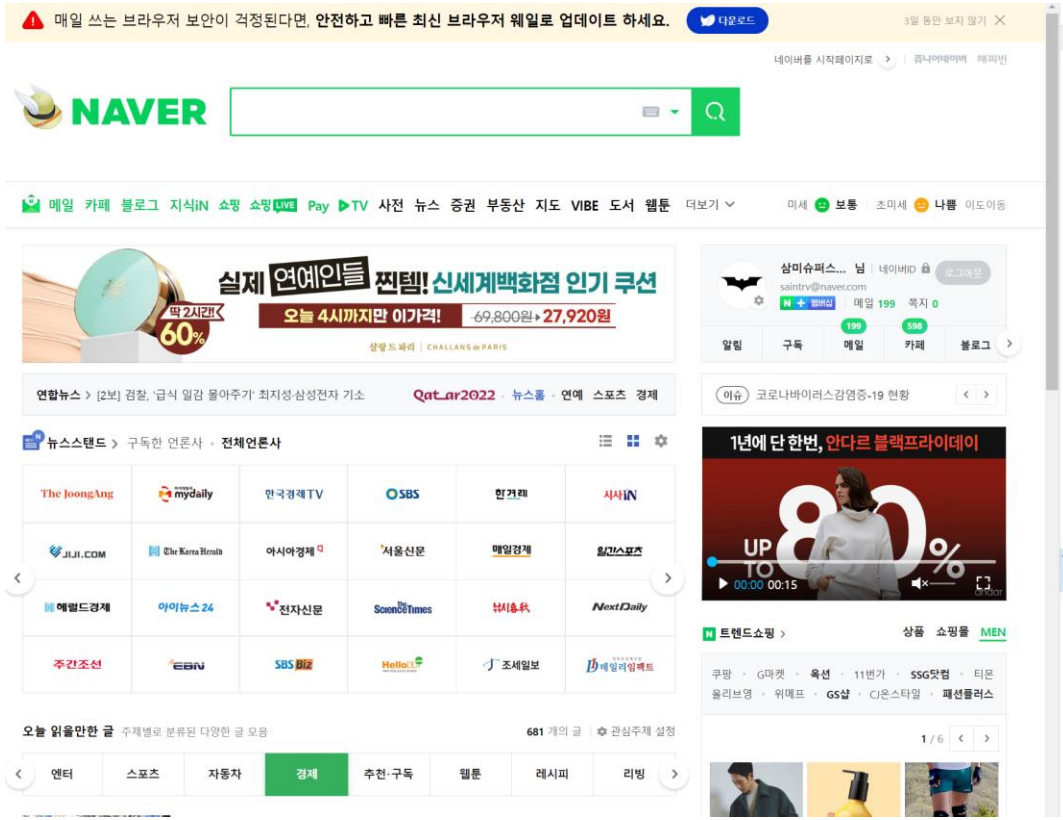
For more details, please see the release notes.

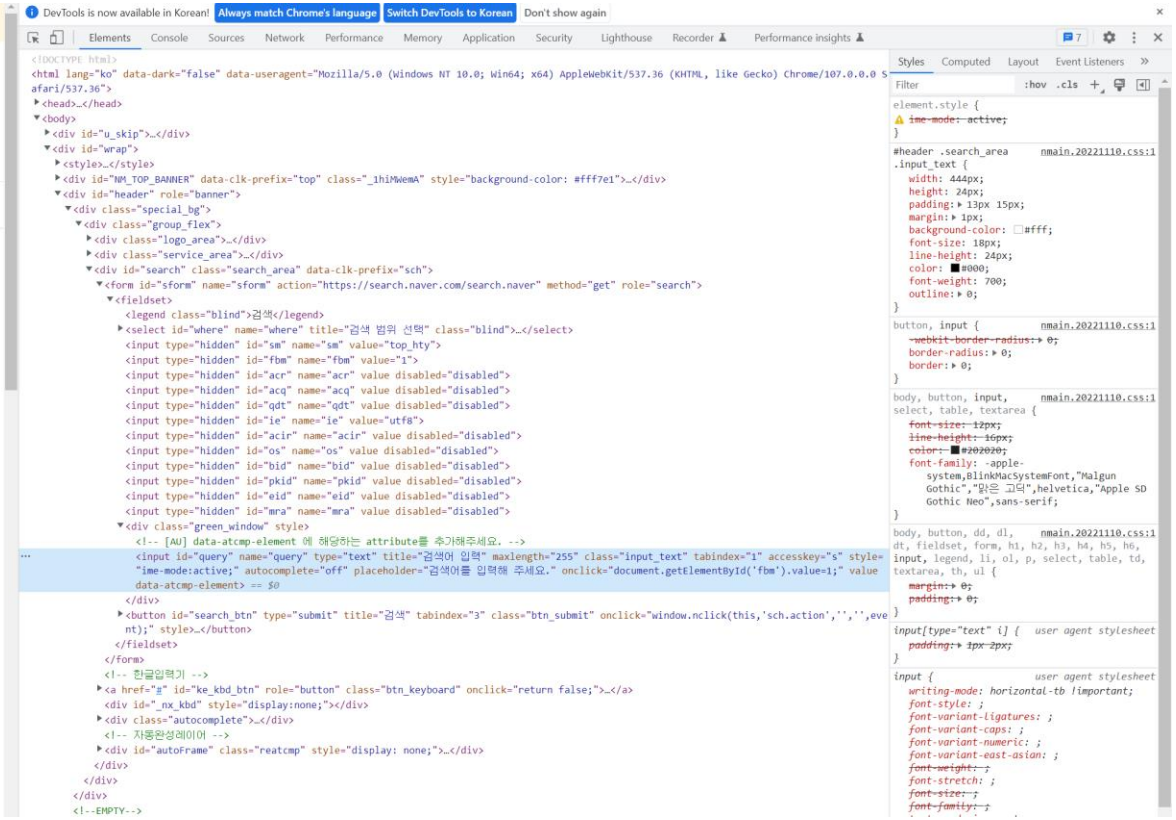


**Index of /94.0.4606.41/**

| Name | Last modified | Size | ETag |
|---|---|---|---|
| Parent Directory | | - | |
| chromedriver_linux64.zip | 2021-09-09 09:57:33 | 9.42MB | 9a00e20a5e5646c009c0ad74a446df10 |
| chromedriver_mac64.zip | 2021-09-09 09:57:36 | 7.80MB | f89f291ccc799a9de32e832807ce7e42 |
| chromedriver_mac64_m1.zip | 2021-09-09 09:57:38 | 7.15MB | 69b92932297c2ce2764aa4c7744e1c17 |
| chromedriver_win32.zip | 2021-09-09 09:57:41 | 5.72MB | c853398ecdb35290e60e57a833e61752 |
| notes.txt | 2021-09-09 09:57:45 | 0.00MB | b88b7fe40b1af6891a9f57da13e08914 |

# Naver Search



# Naver Search

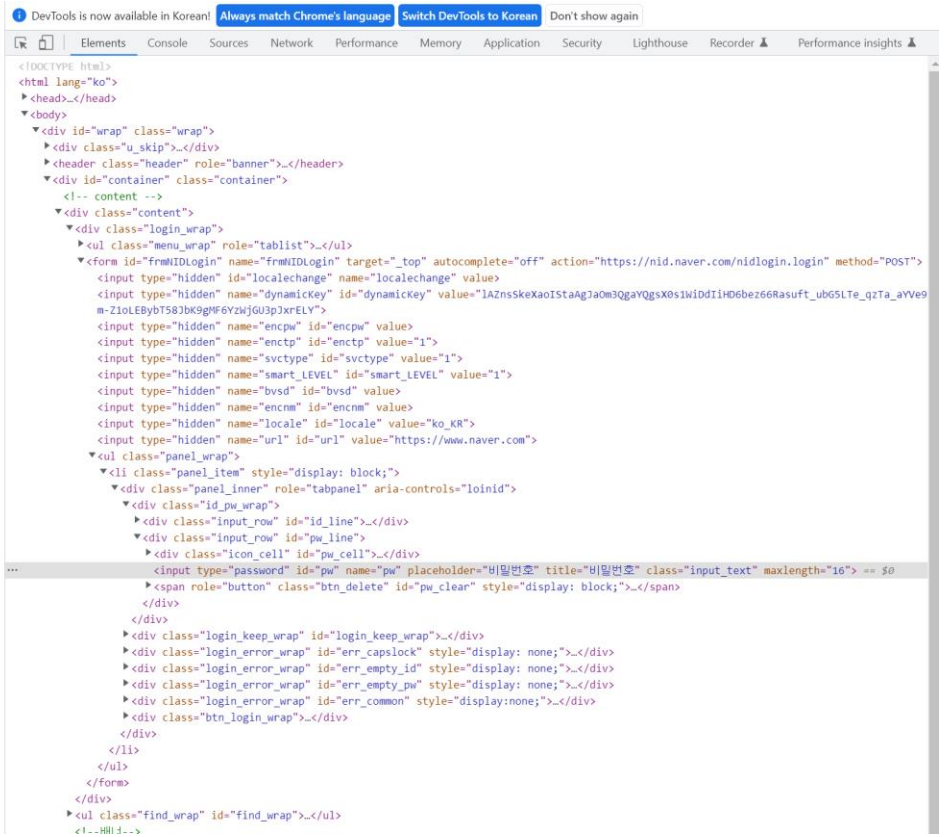
```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

browser = webdriver.Chrome("chromedriver.exe")
browser.get("http://www.naver.com")

#네이버 검색창 (id를 통해 찾거나 class를 통해 찾을 수 있습니다.)
input = browser.find_element(By.CLASS_NAME, "input_text")
input.send_keys("미래에셋대우")
input.submit()
time.sleep(5)
print(browser.page_source)
```

# Chrome version error

```
  File "C:\Users\saint\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\site-packa
ges\selenium\webdriver\remote\webdriver.py", line 444, in execute
    self.error_handler.check_response(response)
  File "C:\Users\saint\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\site-packa
ges\selenium\webdriver\remote\errorhandler.py", line 249, in check_response
    raise exception_class(message, screen, stacktrace)
selenium.common.exceptions.SessionNotCreatedException: Message: session not created: This version of ChromeDriver only supports Chrome version
108
Current browser version is 107.0.5304.107 with binary path C:\Program Files\Google\Chrome\Application\chrome.exe
Stacktrace:
Backtrace:
        (No symbol) [0x00A1E563]
        (No symbol) [0x009A7FC1]
        (No symbol) [0x0089D04D]
        (No symbol) [0x008BF6B4]
        (No symbol) [0x008BA97C]
```

# Naver Login

- Use Chrome Developer Tool to verify the ID of your ID/password



# Naver Login

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
browser = webdriver.Chrome("chromedriver.exe")
browser.get("https://nid.naver.com/nidlogin.login?mode=form&url=https%3A%2F%2Fwww.naver.com")

id = browser.find_element(By.ID, "id")
id.send_keys("myid")

passwd = browser.find_element(By.ID, "pw")
passwd.send_keys("mypassword")
passwd.submit()
time.sleep(5)
```

# Google Image Search

- excute_script
  - Excute java script

```python
# Function for scrolling to the bottom of Google
# Images results
def scroll_to_bottom():

    last_height = driver.execute_script('\
    return document.body.scrollHeight')

    while True:
        driver.execute_script('\
        window.scrollTo(0,document.body.scrollHeight)')

        # waiting for the results to load
        # Increase the sleep time if your internet is slow
        time.sleep(3)

        new_height = driver.execute_script('\
        return document.body.scrollHeight')

        # checking if we have reached the bottom of the page
        if new_height == last_height:
            break

        last_height = new_height
```

# Google Image Search

- Screenshot(save_file_path)
  - Save image of node to save_file_path

```
# Loop to capture and save each image
for i in range(1, 50):

    # range(1, 50) will capture images 1 to 49 of the search results
    # You can change the range as per your need.
    try:
        print(i)
    # XPath of each image
        img = driver.find_element(By.XPATH,
            '//*[@id="islrg"]/div[0]/div[' +
        str(i) + ']/a[1]/div[1]/img')

        # Enter the location of folder in which
        # the images will be saved
        img.screenshot('Download-Location' +
                       query + ' (' + str(i) + ').png')
        # Each new screenshot will automatically
        # have its name updated

        # Just to avoid unwanted errors
        time.sleep(0.2)

    except:

        # if we can't find the XPath of an image,
        # we skip to the next image
        continue
```

# Google Image Search

- Use XPATH to find node

```
# What you enter here will be searched for in
# Google Images
query = "dogs"

# Creating a webdriver instance
driver = webdriver.Chrome("chromedriver.exe")

# Maximize the screen
driver.maximize_window()

# Open Google Images in the browser
driver.get('https://images.google.com/')

# Finding the search box
box = driver.find_element(By.XPATH, '//*[@maxlength="2048"]')

# Type the search query in the search box
box.send_keys(query)

# Pressing enter
box.send_keys(Keys.ENTER)
```

# XPATH

- XPath (short for XML Path) is a query language used for navigating and selecting portions of an XML document.

```
<bib>
   <book price="30">  <publisher> Addison-Wesley </publisher>
              <author> Serge Abiteboul </author>
              <author> <first-name> Rick </first-name>
                          <last-name> Hull </last-name>
              </author>
              <author> Victor Vianu </author>
              <title> Foundations of Databases </title>
              <year> 1995 </year>
   </book>
   <book price="55">
              <publisher> Freeman </publisher>
              <author> Jeffrey D. Ullman </author>
              <title> Principles of Database and Knowledge Base Systems </title>
              <year> 1998 </year>
   </book>

</bib>
```

# Xpath: Summary

| | |
|---|---|
| bib | matches a bib element |
| * | matches any element |
| / | matches the root element |
| /bib | matches a bib element under root |
| bib/paper | matches a paper in bib |
| bib//paper | matches a paper in bib, at any depth |
| //paper | matches a paper at any depth |
| paper\|book | matches a paper or a book |
| @price | matches a price attribute |
| bib/book/@price | matches price attribute in book, in bib |
| bib/book/[@price<“55”]/author/lastname | matches… |

# XPATH

- box = driver.find_element(By.XPATH , '//*[@maxlength="2048"]')

- *Selection of nodes by position*

  `//a[3]` selects the third link in the document

  `//table[last()]` selects the last table in the document

  `//a[position() < 3]` selects the first three links in the document

- *Asterisks (*) match any set of characters or nodes, and can be used in a variety of situations*

  `//table/tr/*` selects all children of `tr` tags in all tables (this is good for selecting cells using both `th` and `td` tags)

  `//div[@*]` selects all `div` tags that have any attributes

- *Root nodes versus non-root nodes*

  `/div` will select the div node only if it is at the root of the document

  `//div` selects all divs anywhere in the document

- *Attribute selection*

  `//@href` selects any nodes with the attribute `href`

  `//a[@href='http://google.com']` selects all links in the document that point to Google

# Google Image Search

- Use XPATH to find node

```
# What you enter here will be searched for in
# Google Images
query = "dogs"

# Creating a webdriver instance
driver = webdriver.Chrome("chromedriver.exe")

# Maximize the screen
driver.maximize_window()

# Open Google Images in the browser
driver.get('https://images.google.com/')

# Finding the search box
box = driver.find_element(By.XPATH, '//*[@maxlength="2048"]')

# Type the search query in the search box
box.send_keys(query)

# Pressing enter
box.send_keys(Keys.ENTER)
```