

In [2]:

```
from bs4 import BeautifulSoup
from selenium import webdriver
from IPython.display import Image

import re
```

```
# Anaconda Prompt 에서 pip install selenium 실행
# !pip install selenium
```

In [3]:

```
driver_path = "driver/chromedriver.exe"
driver = webdriver.Chrome(executable_path=driver_path)
# 가상의 웹 브라우저 띄움
```

In [6]:

```
url_page = 'http://www.seleniumhq.org/projects/webdriver/'
driver.get(url_page)
```

In [7]:

```
driver.find_element_by_css_selector('#menu_download > a').click() #자동클릭
```

In [8]:

```
capture_img = './data/images/seleniumhq_download.png'  
driver.save_screenshot(capture_img)  
Image(capture_img)
```

Out[8]:

Tickets for [SeleniumConf London] are on sale

NOW Go [HERE!](#)

Se

SeleniumHQ

Browser Automation

Projects

Download

Selenium Downloads

Previous Releases

Source Code

Maven Information

Donate to Selenium

with PayPal

Donate


VISA


through sponsorship


You can [sponsor the Selenium project](#) if you'd like some public recognition of your generous contribution.

Selenium Sponsors

See who [supports the Selenium project](#).

 BrowserStack

 SAUCE LABS

 Experitest

New Relic

SYNTHETIC

WEBSITE MONITORING WITH Seleni

Downloads

Below is where you can find the latest Selenium downloads. You can also find a list of [previous](#) information for [Maven users](#) (Maven users only).

Selenium Standalone Server

The Selenium Server is needed in order to run Selenium 3.X is no longer capable of running tests through emulation and the WebDriver interface.

Download version [3.141.59](#)

To run Selenium tests exported from Selenium WebDriver [Runner](#).

To use the Selenium Server in a Grid environment, see [Selenium Grid](#).

The Internet Explorer Driver Setup

This is required if you want to make Selenium WebDriver InternetExplorerDriver work on Windows. Add your \$PATH (or %PATH% on Windows) to the system path expected.

Download version 3.14.0 for (recommended) [Windows IE](#) [CHANGELOG](#)

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with Selenium Remote WebDriver) or create tests that need to make use of language-specific bindings, both 1.x and 2.x style clients.

While language bindings for [other languages](#) are supported by the main project

Language	Client Version	Release Date
Java	3.141.59	2019-08-14
C#	3.14.0	2019-08-14
Ruby	3.14.0	2019-08-14

In [10]:

```
driver.find_element_by_xpath('//*[@id="menu_documentation"]/a').click()
```

## 네이버 크롤링

In [11]:

```
url_page = 'https://naver.com'  
driver.get(url_page)
```

In [1]:

```
# XQuery  
driver.find_element_by_css_selector('#PM_ID_serviceNavi > li:nth-child(6) > a > span.an_icon').click()
```

In [33]:

```
# 평점&리뷰 1등 페이지로 이동  
driver.find_element_by_css_selector('#review1 > div > a').click()
```

In [16]:

```
driver.find_element_by_css_selector('#movieEndTabMenu > li:nth-child(6) > a').click()
```

In [17]:

```
current_url = driver.current_url  
current_url
```

Out[17]:

```
'https://movie.naver.com/movie/bi/mi/review.nhn?code=167560'
```

In [18]:

```
# pattern = re.compile("[Wd+,?]+")
pattern = re.compile("Wd{6}")
result = pattern.findall(current_url)
code = result[0] if len(result)>0 else ''
code
```

Out[18]:

```
'167560'
```

In [19]:

```
review_url = current_url.replace(code, '{}')
review_url
```

Out[19]:

```
'https://movie.naver.com/movie/bi/mi/review.nhn?code={}'
```

In [ ]:

## NAVER 영화 리뷰페이지로 바로 이동

In [22]:

```
# 영화제목 검색기 입력
#movie_title = '봉오동 전투'
movie_title = '알라딘'
driver.find_element_by_css_selector('#ipt_tx_srch').send_keys(movie_title)
```

In [23]:

```
# 검색 클릭
driver.find_element_by_css_selector('#jSearchArea > div > button').click()
```

In [26]:

```
# 검색 첫번째 페이지로 이동
driver.find_element_by_css_selector('#old_content > ul:nth-child(4) > li:nth-child(1) > dl > dt > a').click()
```

In [27]:

```
# 페이지 BeautifulSoup 객체로 가져온다
html = driver.page_source
soup = BeautifulSoup(html, "lxml")
```

In [28]:

```
soup.title
```

Out[28]:

```
<title>알라딘 : 네이버 영화</title>
```

In [29]:

```
current_url = driver.current_url
current_url
```

Out[29]:

```
'https://movie.naver.com/movie/bi/mi/basic.nhn?code=163788'
```

In [30]:

```
pattern = re.compile('Wd{6}')
result = pattern.findall(current_url)
cod = result[0] if len(result) > 0 else ''
code
```

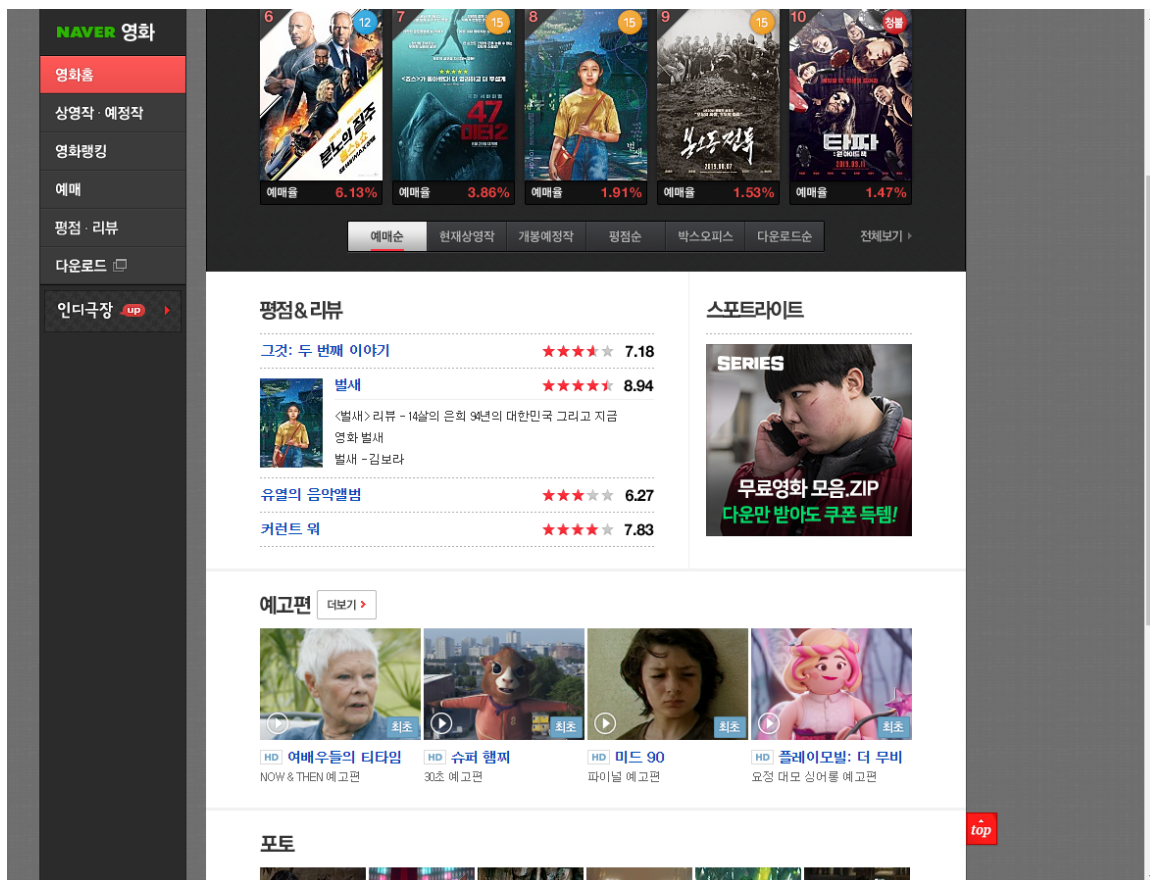
Out[30]:

```
'167560'
```

In [36]:

```
capture_img = './data/images/naver_movie_{code}.png'.format(code=code)
driver.save_screenshot(capture_img)
Image(capture_img)
```

Out[36]:



In [37]:

```
# 리뷰페이지로 바로 이동
url_page = review_url.format(code)
driver.get(url_page)
```

In [40]:

```
html = driver.page_source
soup = BeautifulSoup(html, "lxml")
soup.title
```

Out[40]:

<title>알라딘 : 네이버 영화</title>

In [41]:

```
movie_title = soup.title.get_text().split(':')[0].strip()
movie_title
```

Out[41]:

'알라딘'

In [ ]: