본 자료 및 영상 컨텐츠는 저작권법 제25조 2항에 의해 보호를 받습니다. 본 컨텐츠 및 컨텐츠 일부 문구등을 외부에 공개, 게시하는 것을 금지합니다. 특히 자료에 대해서는 저작권법을 엄격하게 적용하겠습니다.

참고: Scrapy 와 Selenium

Copyright, Dave Lee, 잔재미코딩 (www.fun-coding.org)

Scrapy 와 Selenium

- Scrapy 는 일반적으로는 대량의 정적 웹페이지 크롤링을 위해 활용
- 동적 웹페이지는 지원하지 않으므로, Selenium 과 연계해서 활용하고자 하는 니즈가 있음
- 안정성과 성능을 고려했을 때, 두 기술을 조합하는 것을 추천하지는 않음
    - 즉 동적 웹페이지는 Selenium 자체의 코드로만 작성하는 것을 추천드림
- 단, 관련 니즈가 있었기 때문에, 가볍게 참고 차원에서 공유드리는 것임

Copyright, Dave Lee, 잔재미코딩 (www.fun-coding.org)

Scrapy와 Selenium

- 사전 작업

```
> scrapy startproject myselenium
> cd myselenium
> scrapy genspider selenium_test davelee-fun.github.io/blog/TEST/index.html
```

Copyright, Dave Lee, 잔재미코딩 (www.fun-coding.org)

Scrapy와 Selenium: spider 코드 작성 (2022.06.30 업데이트)

> __init__() 에서 chromedriver 로 self.driver 정의 후, parse() 에서 selenium 코드를 작성하면 됨

```
import scrapy
from selenium import webdriver
import time
# [2022.06.30] find_element_by_() 함수는 find_element(By., ) 과 같은 형태로 함수가 변경됨에 따른 추가 코드
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager

class SeleniumTestSpider(scrapy.Spider):
    name = 'selenium_test'
    allowed_domains = ['davelee-fun.github.io/blog/TEST/index.html']
    start_urls = ['https://davelee-fun.github.io/blog/TEST/index.html']

    def __init__(self):
        # chromedriver = 'C:/dev_python/Webdriver/chromedriver.exe' # 윈도우
        chromedriver = '/usr/local/Cellar/chromedriver/chromedriver'  # 맥
        headlessoptions = webdriver.ChromeOptions()
        headlessoptions.add_argument('headless')
        self.driver = webdriver.Chrome(service=Service(chromedriver), options=headlessoptions)

    def parse(self, response):
        self.driver.get(response.url)
        time.sleep(2)
        elem = self.driver.find_elements(By.CSS_SELECTOR, ".news")
        print(elem.text)
        pass
```

- start_urls 에 자동으로 http:// 와 마지막에 / 가 들어갈 수 있으므로, https:// 로 변경 및, 마지막의 / 문자는 삭제 필요

Copyright, Dave Lee, 잔재미코딩 (www.fun-coding.org)