



박소희 <parksohee63@gmail.com>

[프로젝트] 3. 주식을 줍줍하자

1 개의 메일

박소희 <parksohee63@gmail.com>

2020년 8월 13일 오전 2:55

받는사람: 박소희 <parksohee63@gmail.com>

5% 이상 오른 주식에 대해 알려주는 서비스입니다. 이번 시간을 통해 다음 내용을 배울 수 있습니다.

1. 브라우저 자동화를 통해 네이버 주식 정보 가져오기
2. 브라우저 자동화할 때 보이지 않도록 만들기 (백그라운드 작동)
3. 프로그램이 반복실행되도록 만들기 (10초마다, 5분마다, 매일, 매주, ..etc)

(업무자동화에 활용할 수 있겠죠?)

pycharm의 new project를 클릭해서, **sparta** → **project** → **stockinfo** 폴더를 만들고 시작하기!

- 1. 패키지 설치하기 & 폴더 세팅하기

설치할 패키지: bs4 selenium schedule

static, templates 폴더 + [app.py](#) 이제 너무 익숙하죠?

- 2. [브라우저 자동화] 네이버 주식 정보 가져오기

1. 브라우저 동작을 자동화하고 싶다면?

지금까지는 단순 정보 조회만을 자동화했습니다. 그런데 단순히 정보를 가져오는 것을 넘어서, 다음 동작까지 하게 하려면 어떻게 해야 할까요?


- 아이디와 비밀번호를 자동으로 입력하고 클릭까지 하게 하려면?
- 클릭으로 파일 다운로드까지 자동으로 하게 하려면?
- 내가 원하는 시간에 영화 표 예매까지 자동으로 하게 하려면?

이럴 때 필요한 녀석이 바로 Selenium(셀레니움)입니다!


2. 브라우저 자동화를 위해선 'chromedriver'라는 것이 필요해요! ([다운로드 링크](#))

크롬을 자동화 시켜주는 친구죠. 내 크롬 버전에 맞게 다운로드 받아볼까요?

- 내 크롬 버전을 확인하기

 <https://s3-us-west-2.amazonaws.com/secure.notion-static.com/577b730f-6d96-421d-88ed-6f00d98cde8c/Untitled.png>

- 버전에 맞는 드라이버 다운로드 하기

 <https://s3-us-west-2.amazonaws.com/secure.notion-static.com/acfe48bc-66ad-4b53-810c-fb30d899e0d7/Untitled.png>

3. 기본 코드 - 여기서 출발하세요!

```
from selenium import webdriver
from bs4 import BeautifulSoup
```

```
# 내장 라이브러리로 설치할 필요가 없습니다.
import time
```

```
# 셀레니움을 실행하는데 필요한 크롬드라이버 파일을 가져옵니다.
driver = webdriver.Chrome('chromedriver')
```

```
# 네이버 주식페이지 url을 입력합니다.
url = '<a href="https://m.stock.naver.com/item/main.nhn#/stocks/005930/total">https://m.stock.naver.com/item/main.nhn#/stocks/005930/total</a>'
```

```
# 크롬을 통해 네이버 주식페이지에 접속합니다.
driver.get(url)
```

```
# 크롬을 종료합니다.
driver.quit()
```

4. 완성 코드

```
from selenium import webdriver
from bs4 import BeautifulSoup

import time

# 셀레니움을 실행하는데 필요한 크롬드라이버 파일을 가져옵니다.
driver = webdriver.Chrome('chromedriver')

# 네이버 주식페이지 url을 입력합니다.
url = '<https://m.stock.naver.com/item/main.nhn#/stocks/005930/total>'

# 크롬을 통해 네이버 주식페이지에 접속합니다.
driver.get(url)

# 정보를 받아오기까지 2초를 잠시 기다립니다.
time.sleep(2)

# 크롬에서 HTML 정보를 가져오고 BeautifulSoup을 통해 검색하기 쉽도록 가공합니다.
soup = BeautifulSoup(driver.page_source, 'html.parser')

name = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.item_wrp > div > h2').text

current_price = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > strong').text

rate = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > div > span.gap_rate > span.rate').text

print(name,current_price,rate)

# 크롬을 종료합니다.
driver.quit()
```

- 3. [브라우저 자동화] 여러 주식 정보 가져오기 + 브라우저 안보이게

1. 2번을 여러번 돌리면 되겠죠?

```
#삼성전자, 네이버, SK텔레콤, SK이노베이션, 카카오
codes = ['005930', '035420', '017670', '096770', '035720']

from selenium import webdriver
from bs4 import BeautifulSoup

import time

# 셀레니움을 실행하는데 필요한 크롬드라이버 파일을 가져옵니다.
driver = webdriver.Chrome('chromedriver')

codes = ['005930', '035420', '017670', '096770', '035720']

for code in codes:

    # 네이버 주식페이지 url을 입력합니다.
    url = '<https://m.stock.naver.com/item/main.nhn#/stocks/'+code+'/total>'

    # 크롬을 통해 네이버 주식페이지에 접속합니다.
    driver.get(url)

    # 정보를 받아오기까지 2초를 잠시 기다립니다.
    time.sleep(2)

    # 크롬에서 HTML 정보를 가져오고 BeautifulSoup을 통해 검색하기 쉽도록 가공합니다.
    soup = BeautifulSoup(driver.page_source, 'html.parser')

    name = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp >
```

```
div > div:nth-child(2) > div > div.item_wrp > div > h2').text

current_price = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_
info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > strong').text

rate = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp >
div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > div > span.gap_rate >
span.rate').text

print(name,current_price,rate)

# 크롬을 종료합니다.
driver.quit()
```

2. 브라우저를 안 뜨게 하려면?

#셀레니움의 option 값을 조절하면 돼요! 이렇게

```
options = webdriver.ChromeOptions()
options.add_argument('headless')
options.add_argument('window-size=1920x1080')
options.add_argument("disable-gpu")
options.add_argument("user-agent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36")

driver = webdriver.Chrome('chromedriver',options=options)
```

3. 완성코드

```
from selenium import webdriver
from bs4 import BeautifulSoup

import time

### option 적용 ###
options = webdriver.ChromeOptions()
options.add_argument('headless')
options.add_argument('window-size=1920x1080')
options.add_argument("disable-gpu")
options.add_argument("user-agent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36")

driver = webdriver.Chrome('chromedriver',options=options)
#####

codes = ['005930', '035420', '017670', '096770', '035720']

for code in codes:

    # 네이버 주식페이지 url을 입력합니다.
    url = '<https://m.stock.naver.com/item/main.nhn#/stocks/'+code+'/total>'

    # 크롬을 통해 네이버 주식페이지에 접속합니다.
    driver.get(url)

    # 정보를 받아오기까지 2초를 잠시 기다립니다.
    time.sleep(2)

    # 크롬에서 HTML 정보를 가져오고 BeautifulSoup을 통해 검색하기 쉽도록 가공합니다.
    soup = BeautifulSoup(driver.page_source, 'html.parser')

    name = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp >
div > div:nth-child(2) > div > div.item_wrp > div > h2').text

    current_price = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_
info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > strong').text

    rate = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp >
div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > div > span.gap_rate >
span.rate').text

    print(name,current_price,rate)
```

```
# 크롬을 종료합니다.
driver.quit()
```

• 4. [반복 실행] 5분에 한번씩 실행하게 하고 싶다면?

1. 파이썬 파일이 5분마다 실행되는 걸까요?

→ 아니요! 파이썬 파일은 계속 실행되고 있고, 5분에 한번씩 해당 '함수'가 불리는 거예요 → 즉, 5분에 한번씩 같은 동작을 수행하는 것이죠

2. 기본 코드 - 여기서 출발하세요

```
import schedule

def job():
    print('여기에 할 일을 넣기')

def run():
    schedule.every(1).seconds.do(job) #1초에 한번씩 실행 (5분은 너무 길어서..^^;)
    while True:
        schedule.run_pending()

if __name__ == "__main__":
    run()
```

3. 완성코드: 주식 코드를 job() 안으로 넣기

```
import schedule

from selenium import webdriver
from bs4 import BeautifulSoup

import time

def get_my_stock():
    ### option 적용 ###
    options = webdriver.ChromeOptions()
    options.add_argument('headless')
    options.add_argument('window-size=1920x1080')
    options.add_argument("disable-gpu")
    options.add_argument(
        "user-agent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36")

    driver = webdriver.Chrome('chromedriver', options=options)
    #####

    codes = ['005930', '035420', '017670', '096770', '035720']

    for code in codes:
        # 네이버 주식페이지 url을 입력합니다.
        url = '<a href="https://m.stock.naver.com/item/main.nhn#/stocks/">https://m.stock.naver.com/item/main.nhn#/stocks/</a>' + code + '/total'

        # 크롬을 통해 네이버 주식페이지에 접속합니다.
        driver.get(url)

        # 정보를 받아오기까지 2초를 잠시 기다립니다.
        time.sleep(2)

        # 크롬에서 HTML 정보를 가져오고 BeautifulSoup을 통해 검색하기 쉽도록 가공합니다.
        soup = BeautifulSoup(driver.page_source, 'html.parser')

        name = soup.select_one(
            '#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.item_wrp > div > h2').text

        current_price = soup.select_one(
            '#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > strong').text

        rate = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > div > span.gap_rate > span.rate').text
```

```

        print(name,current_price,rate)

    print('-----')
    # 크롬을 종료합니다.
    driver.quit()

def job():
    get_my_stock()

def run():
    schedule.every(15).seconds.do(job) #15초에 한번씩 실행
    while True:
        schedule.run_pending()

if __name__ == "__main__":
    run()

```

• 5. [메일보내기] G메일 보내기

1. 파이썬으로 메일도 보내나요?!

네, 할 수 있습니다! 이처럼 누구나 많이 쓰는 동작들은 대부분 라이브러리(미리 짜여진 작은 프로그램)로 작성되어 있으니, 앞으로 여러분이 필요한 것이 있다면 구글에 검색해보세요! 거의 대부분 존재할 거예요




2. 사전 작업


- G메일의 앱 비밀번호를 만들고, 받아오기

1. 일단, google 계정 → 보안에 접속합니다. ([링크](#))


2. 2단계 인증을 설정하고, 앱 비밀번호를 클릭합니다.

 <https://s3-us-west-2.amazonaws.com/secure.notion-static.com/53833fd0-72e4-4850-9046-75f75cf94ec6/Untitled.png>

3. 앱 비밀번호를 생성합니다.

 <https://s3-us-west-2.amazonaws.com/secure.notion-static.com/cc5efc0f-3999-4d85-b710-09605d0c50bf/Untitled.png>

4. 복사해서 적어두세요! 곧 이용할 예정!

 <https://s3-us-west-2.amazonaws.com/secure.notion-static.com/eea8b75a-87f7-45a9-a2ea-888a57fdc63b/Untitled.png>

3. 기본 코드 (내용은 길지만, 별로 어렵지 않아요~)

```

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

# 내 이메일 정보를 입력합니다.
me = "test1@abc.com"
# 내 비밀번호를 입력합니다.
my_password = "wpfzfpqiubxjnjun"
# 이메일 받을 상대방의 주소를 입력합니다.
you = "test2@abc.com"

## 여기서부터 코드를 작성하세요.
# 이메일 작성 form을 받아옵니다.
msg = MIMEMultipart('alternative')
# 제목을 입력합니다.
msg['Subject'] = "알림!"
# 송신자를 입력합니다.
msg['From'] = me
# 수신자를 입력합니다.
msg['To'] = you

# 이메일 내용을 작성합니다.
html = '이렇게 알리를 줄 수 있죠!'

```

```
# 이메일 내용의 타입을 지정합니다.
part2 = MIMEText(html, 'html')
# 이메일 form에 작성 내용을 입력합니다
msg.attach(part2)
## 여기에서 코드 작성이 끝납니다.

# Gmail을 통해 전달할 것임을 표시합니다.
s = smtplib.SMTP_SSL('smtp.gmail.com')
# 계정 정보를 이용해 로그인합니다.
s.login(me, my_password)
# 이메일을 발송합니다.
s.sendmail(me, you, msg.as_string())
# 이메일 보내기 프로그램을 종료합니다.
s.quit()
```

4. 완성코드: 15초 마다 돌면서 조건에 맞으면 메일로 알려주기

조건은, 전일 대비 4% 이상 오르면 - 으로 잡아줬어요!

```
import schedule

from selenium import webdriver
from bs4 import BeautifulSoup

import time

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

def send_mail(stock_name):
    # 내 이메일 정보를 입력합니다.
    me = "test1@abc.com"
    # 내 비밀번호를 입력합니다.
    my_password = "wpfzfpqiubxjnjun"
    # 이메일 받을 상대방의 주소를 입력합니다.
    you = "test2@abc.com"

    ## 여기서부터 코드를 작성하세요.
    # 이메일 작성 form을 받아옵니다.
    msg = MIMEMultipart('alternative')
    # 제목을 입력합니다.
    msg['Subject'] = "알림!"
    # 송신자를 입력합니다.
    msg['From'] = me
    # 수신자를 입력합니다.
    msg['To'] = you

    # 이메일 내용을 작성합니다.
    html = stock_name+' 주식을 한번 보세요!'
    # 이메일 내용의 타입을 지정합니다.
    part2 = MIMEText(html, 'html')
    # 이메일 form에 작성 내용을 입력합니다
    msg.attach(part2)
    ## 여기에서 코드 작성이 끝납니다.

    # Gmail을 통해 전달할 것임을 표시합니다.
    s = smtplib.SMTP_SSL('smtp.gmail.com')
    # 계정 정보를 이용해 로그인합니다.
    s.login(me, my_password)
    # 이메일을 발송합니다.
    s.sendmail(me, you, msg.as_string())
    # 이메일 보내기 프로그램을 종료합니다.
    s.quit()

def get_my_stock():
    ### option 적용 ###
    options = webdriver.ChromeOptions()
    options.add_argument('headless')
    options.add_argument('window-size=1920x1080')
    options.add_argument("disable-gpu")
    options.add_argument(
        "user-agent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36")
```

```

driver = webdriver.Chrome('chromedriver', options=options)
#####

codes = ['005930', '035420', '017670', '096770', '035720']

for code in codes:
    # 네이버 주식페이지 url을 입력합니다.
    url = '<a href="https://m.stock.naver.com/item/main.nhn#/stocks/">https://m.stock.naver.com/item/main.nhn#/stocks/>' + code + '/total'

    # 크롬을 통해 네이버 주식페이지에 접속합니다.
    driver.get(url)

    # 정보를 받아오기까지 2초를 잠시 기다립니다.
    time.sleep(2)

    # 크롬에서 HTML 정보를 가져오고 BeautifulSoup을 통해 검색하기 쉽도록 가공합니다.
    soup = BeautifulSoup(driver.page_source, 'html.parser')

    name = soup.select_one(
        '#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.item_wrp > div > h2').text

    current_price = soup.select_one(
        '#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > strong').text

    rate = soup.select_one('#header > div.end_header_topinfo > div.flick-container.major_info_wrp > div > div:nth-child(2) > div > div.stock_wrp > div.price_wrp > div > span.gap_rate > span.rate').text

    print(name,current_price,rate)

    if (float(rate) > 4):
        print('send',name)
        send_mail(name)

print('-----')
# 크롬을 종료합니다.
driver.quit()

def job():
    get_my_stock()

def run():
    schedule.every(10).seconds.do(job) #10초에 한번씩 실행
    while True:
        schedule.run_pending()

if __name__ == "__main__":
    run()

```