[스파르타] 파이썬 혼자놀기 패키지

코로나19 물럿거라! - 집에서 심심한 그대여, 파이썬을 내 장난감으로 만들자!

▼ **(←누르기) 본격적인 학습 시작 전! 한번 눌러보세요.**

- 펼쳐졌습니다! 잘하셨어요. 다시 한 번 누르면 닫힙니다.
  앞으로 만날 때만다 꼭 한번씩 펼쳐서 내용을 확인해주세요 😊

[들어가기에 앞서]

🚨 본 강의는 코딩 왕초보분들이 빠르고 쉽게 파이썬을 배우고, 응용 하는 것을 목표로 합니다.
웹서비스 개발의 전 과정이 궁금하신 분들은 "스파르타코딩클럽" 정규수업을 참고하세요!

[공부하는 방법] : 휴대폰으로 영상을 켜두세요!(링크)

✍️ **휴대폰에 → 영상**을 켜두고, **PC에 → 강의자료와 파이참**을 띄워두고 하면 편해요!

[질문도 하세요!] : 웹페이지 오른쪽 아래 톡버튼(링크)을 통해서!

🤔 집에서 공부 좀 하려는데, 막히면 안되죠! 사회적 거리두기 종료까지(~04/19) 3명의 튜터가 붙어서 질문에 답 드리기로 하였습니다. 마음껏 물어봐주세요. 최대한 빠르게 답 드릴게요!

[15분] : 사전준비 - 설치하기

👻 "막힘 없는 설치. 한번 가봅시다!"

▼ 1) 파이썬 & 파이참 설치하기

💻 Mac OS

🖥️ Windows OS

[1시간 30분] : 파이썬 기초 문법 익히기

👻 "주어, 동사, 목적어 다 배우고 I Love You 하면 지쳐요. 필요한 것만 배웁시다!"

▼ 2) 프로그래밍 언어란?

- 컴퓨터와 대화하는 방법입니다.

  👉 컴퓨터는 0, 1밖에 알아듣지 못해서 101010001 과 같이 써주어야 합니다. 하지만 이런 건 사람이 이렇게 쓰기는 어렵죠! 그래서 보다 쉽게 컴퓨터에게 명령을 내릴 수 있도록 뛰어난 개발자들이 사람이 읽고 쓰기 편한 프로그래밍 언어를 만들어 두었고, 우리는 그걸 사용하면 되겠습니다.

- 먹다 = Eat = 吃 = ...

  👉 프로그래밍 언어는 Python, C, C++, Java 등 다양한 종류가 존재합니다. 근본적인 능력 차이는 없지만 각자의 특장점이 조금씩 다르다고 할 수 있는데요.

  예를 들어 한국어로는 '누리끼리하다', '노랗다', '개나리색' 등의 표현이 어릴적부터 가능하지만, 영어로는 'yellow' 하나밖에 기억나지 않는 것처럼요.

  그치만 성인이 되고 언어의 끝판왕이 되면 모든 표현이 가능하듯이, 프로그래밍 언어도 '고수'가 되면 큰 차이가 없습니다!

- 뭘 할 수 있나요?

  👉 컴퓨터로 우리가 했던 모든 것을 할 수 있어요. 예를 들면 1시간 뒤에 컴퓨터를 꺼지게 할 수도 있고, 브라우저를 켜서 정보를 가져오게 할 수도 있고, 가져온 정보를 엑셀에 저장하게 만들 수도 있죠.

▼ 3) 파이썬을 설치한다는 것의 의미 / 파이참은 무엇일까?

- 파이썬을 설치한다?

  👉 파이썬을 설치한다는 것: 일종의 번역팩을 설치한다고 생각하면 됩니다. 컴퓨터는 101010001 과 같은 언어만 알아듣는다고 했지요? 파이썬 문법으로 된 것을 101010001로 변환해줄 수 있도록, 번역 패키지를 설치하는 것입니다.

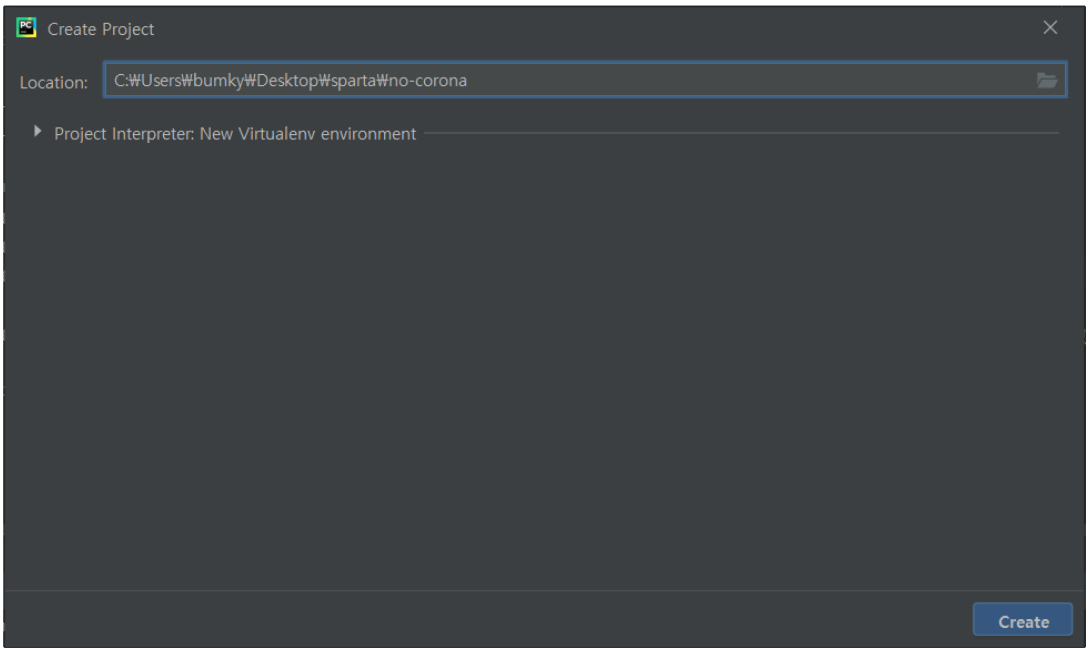
- 파이참은 무엇일까?

  👉 PyCharm은 파이썬을 이용해 코딩하는데 유용한 프로그램입니다. 메모장에서도 한글을 칠 수 있지만, 워드로 작성하면 더 편한 것과 같은 맥락입니다. 이런 친구들을 IDE(Integrated development environment, 통합개발환경)라고 합니다.

▼ 4) 파이참 실행하기 & 첫 파이썬 파일 실행

- 파이참 실행하기

> 👉 바탕화면에 폴더를 만들어서 진행하면 편해요! (폴더 이름은 영어를 추천!)



- 파이썬 파일을 하나 만들어 실행해보기

> 👉 hello.py라는 파일에 아래와 같이 적어봅니다.

```
print ('hello world!')
```

Python

▼ 5) 파이썬 문법을 시작하기에 앞서..

- 파이썬은 매우 직관적인 언어이고, 할 수 있는 것도 많습니다. 그런데, 개발자들도 모든 문법을 기억하기란 쉽지 않습니다. 오늘 배우는 것 외에 필요한 것들은 구글링해서 찾아보면 됩니다!
- 본 강의에서 class, tuple, set과 같은 개념을 배우지는 않습니다. 본격적인 개발자로 전향하는데엔 알아둬야 하는 것들이지만, '갖고놀기' 레벨을 달성하는데엔 몰라도 괜찮습니다.

- ▼ 6) 파이썬 문법 - 5가지만 알면 끝!
  - ▶ 변수 & 기본연산
  - ▶ 자료형
  - ▶ 함수
  - ▶ 조건문
  - ▶ 반복문

▼ 7) 파이썬 연습 - 공공데이터를 이용해서 마스크 재고가 남은 약국을 찾아보기

1. OpenAPI란?

> 👉 API (Application Programming Interface) ?
>
> 은행에도 여러가지 데이터(잔고, 금리, 직원명부, 닫는시간 등)가 있지만, 일을 효율적으로 처리하기 위해 '창구'를 두어서 고객을 맞이하죠! 예를 들어, 입출금을 하고 싶다면 '입출금 창구'로 가서 정해진 대로 주민등록증과 입/출금 여부, 금액을 알려주고요.
>
> API는 서버가 일을 처리해주기 위해 만들어둔 '창구'를 의미합니다. 미리 정해진 규칙대로 요청하면, 원하는 데이터를 가져갈 수 있지요.
>
> OpenAPI는, '누구나 요청할 수 있는 창구'로 생각해주시면 됩니다!

2. OpenAPI는 어떻게 사용하나요?

아래 주소를 chrome 주소창에 붙여넣어 볼까요?

```
https://8oi9s0nnth.apigw.ntruss.com/corona19-masks/v1/storesByAddr/json?address=서울특별시 용산구
```

Python

아래와 같은 내용이 보이실 겁니다. 👏용산구의 마스크 약국 데이터👏를 받아오는데 성공하셨어요! 이런 데이터 형식을 JSON이라고 합니다. 마치 우리가 배웠던 '딕셔너리' 자료형과 유사하게 생겼죠?


```
{
  "address": "서울특별시 용산구",
  "count": 127,
  "stores": [{
    "addr": "서울특별시 용산구 후암로57길 37 1층 (동자동)",
    "code": "11801778",
    "created_at": "2020/04/06 10:40:00",
    "lat": 37.5518055,
    "lng": 126.9741258,
    "name": "새평화약국",
    "remain_stat": "plenty",
    "stock_at": "2020/04/06 09:48:00",
    "type": "01"
```


3. 조금 더 예쁘게 보자! - JSONView

(링크)를 클릭해서 JSONView 설치 페이지로 이동한 후, 'Chrome에 추가' > '확장 프

로그램 추가'를 차례대로 클릭해주세요. 그리고 아래 주소를 다시 한 번 chrome 주소

```
https://8oi9s0nnth.apigw.ntruss.com/corona19-masks/v1/storesByAddr/json?address=서울특별시 용산구
```

Python

그러면 아래와 같이 조금 더 보기 편하게 바뀌어 있을 거예요.


```
{
    address: "서울특별시 용산구",
    count: 127,
  - stores: [
      - {
            addr: "서울특별시 용산구 후암로57길 37 1층 (동자동)",
            code: "11801778",
            created_at: "2020/04/06 10:45:00",
            lat: 37.5518055,
            lng: 126.9741258,
            name: "새평화약국",
            remain_stat: "plenty",
            stock_at: "2020/04/06 09:48:00",
            type: "01"
```


4. 파이썬의 강력한 도구! 라이브러리를 설치해보아요

> 👉 파이썬은 라이브러리가 많은 언어로 유명해요! 덕분에 많은 코드를 직접 짜지 않아도 된됩니다. 필요한 게 있으면, google에 미리 검색해보는 것을 추천드릴게요!

```
설치 할 라이브러리: requests
```

Python

▼ 설치 버튼찾기 - Windows

File > Settings > 검색창에 interpreter 입력 > 우측 상단 '플러스'(+) 아이콘 클릭 > 라이브러리 이름 입력 > Install packages 클릭!

▼ 설치 버튼찾기 - Mac

PyCharm > Preference > 검색창에 interpreter 입력 > 하단 좌측 '플러스'(+) 아이콘 클릭 > 라이브러리 이름 입력 > Install packages 클릭!

5. **기본코드: OpenAPI를 이용할 때는 이 코드를 복붙하고 시작하세요!**

```
import requests # 정보를 긁어오는데 필요한 requests 라이브러리를 가져옵니다. # 입력한 주소로 가서 정보를 가져옵니다. r = requests.get('여기에 OpenAPI 주소를 넣으면 됩니다!') # 가져온 정보를 파이썬이 사용할 수 있도록 변경합니다. rjson = r.json()
```

Python

6. 완성본

```
import requests # 정보를 긁어오는데 필요한 requests 라이브러리를 가져옵니다. # 입력한 주소로 가서 정보를 가져옵니다. r = requests.get('https://8oi9s0nnth.apigw.ntruss.com/corona19-masks/v1/storesByAddr/json?address=서울특별시 용산구') # 가져온 정보를 파이썬이 사용할 수 있도록 변경합니다. rjson = r.json() # 상점 정보를 꺼냅니다. store_list = rjson['stores'] # 상점 목록을 돌면서 하나하나 작업을 진행합니다. for store in store_list: # 만약에 충분한 양의 마스크가 남아있는 약국이면 if store['remain_stat'] == 'plenty': # 약국의 주소와 이름을 출력합니다. print(store['addr'],store['name'])
```

Python

7. 다른 구 정보도 가져와봅시다!

👉 에러 핸들링을 위한 try - except 문을 처음 만났군요!

```
import requests # 정보를 긁어오는데 필요한 requests 라이브러리를 가져옵니다. # 원하는 서울시 구 이름을 입력합니다. gus = ['용산구','마포구','종로구','동작구'] # 구 이름마다 다음 내용을 반복 작업합니다. for gu in gus: # '서울특별시 용산구' 또는 '서울특별시 마포구'와 같이 만듭니다. place = '서울특별시 '+gu # 구 이름을 붙여 정보를 요청할 API 주소를 완성합니다. url = 'https://8oi9s0nnth.apigw.ntruss.com/corona19-masks/v1/storesByAddr/json?address='+place # OpenAPI 에 정보를 요청합니다. r = requests.get(url) rjson = r.json() # 상점 정보를 꺼냅니다. store_list = rjson['stores'] for store in store_list: try: # 만약에 충분한 양의 마스크가 남아있는 약국이면 if store['remain_stat'] == 'plenty': # 약국의 주소와 이름을 출력합니다. print(store['addr']+' - '+store['name']) except: # 만약 실행 중 에러가 발생하면 건너뛰고 실행을 계속합니다. continue
```

Python

▼ 8) 문법이 어렵다고요?

- 첫 술에 배부를 수 없듯이 파이썬도 익숙해지는 시간이 필요합니다. 그때 그때 필요할 때마다 검색하고 익혀나간다는 마음을 가져보는 건 어떨까요?
- 파이썬 문법이 더 궁금하신 분들만! → (클릭)

[1시간] : 연습 겸 미니프로젝트

👻 "연습 게임을 먼저 해볼까요? 복사+붙여넣기를 두려워 마세요!"

▼ 9) [크롤링] 지니뮤직 곡 순위를 긁어오기

1. API가 제공되지 않는다면, 직접 가져와볼게요!

> 👉 모든 정보를 API 로 제공하는 건 아니랍니다. 이럴 경우에는 우리가 직접 가져올 수도 있는데요. 이렇게 인터넷에 제공되는 정보를 긁어오는 작업을 '크롤링'이라고 합니다.

2. 기본 코드: 크롤링은 다음 코드에서 출발해주세요.

```
설치 할 라이브러리: bs4
```

Python

```
import requests from bs4 import BeautifulSoup # URL을 읽어서 HTML를 받아오고, headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'} data = requests.get('원하는 페이지 주소를 입력하세요.',headers=headers) # HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만듦 soup = BeautifulSoup(data.text, 'html.parser')
```

Python

3. 완성 코드

```
import requests from bs4 import BeautifulSoup # URL을 읽어서 HTML를 받아오고, headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'} data = requests.get('https://www.genie.co.kr/chart/top200?ditc=D&ymd=20200403&hh=23&rtm=N&pg=1',headers=headers) # HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만듦 soup = BeautifulSoup(data.text, 'html.parser') # select를 이용해서, tr들을 불러오기 songs = soup.select('#body-content > div.newest-list > div > table > tbody > tr') # movies (tr들) 의 반복문을 돌리기 for song in songs: # td 태그 중 클래스가 number인 녀석의 텍스트 중 0, 1번째 글자를 가져오고 좌우 공백을 제거합니다. rank = song.select_one('td.number').text[0:2].strip() # td 태그 중 클래스가 info인 녀석 바로 아래, 클래스가 title, ellipsis 중복으로 붙은 a 태그를 가져옵니다. # 그리고 a 태그의 텍스트 전부를 가져오고 좌우 공백을 제거합니다. title = song.select_one('td.info > a.title.ellipsis').text.strip() # td 태그 중 클래스가 info인 녀석 바로 아래, 클래스가 artist, ellipsis 중복으로 붙은 a 태그를 가져옵니다. # 그리고 a 태그의 텍스트 전부를 가져옵니다. artist =
```

```
song.select_one('td.info > a.artist.ellipsis').text
print(rank,title,artist)
```

Python

▼ 10) [엑셀저장] 지니뮤직 곡 순위를 엑셀에 저장하기

▶ 엑셀파일을 다운로드 받아서, 파이썬 파일 폴더에 둡니다.

1. 가져온 정보를 엑셀에 저장해봅시다!

```
설치 할 라이브러리: openpyxl
```

Python

```
from openpyxl import load_workbook # data_only=True로 해줘야 수식이 아닌 값으로 받아온다. load_wb = load_workbook("mymusic.xlsx", data_only=True) # 시트 이름으로 불러오기 load_ws = load_wb['Sheet1'] # 셀 좌표로 값 출력 print(load_ws.cell(1, 1).value) # 셀 좌표로 값 입력 load_ws.cell(5,7,'5행7열') load_wb.save("mymusic.xlsx")
```

Python

2. 지니뮤직 크롤링과 함께 사용해볼까요?

```
import requests from bs4 import BeautifulSoup from openpyxl import load_workbook # data_only=True로 해줘야 수식이 아닌 값으로 받아온다. load_wb = load_workbook("mystock.xlsx", data_only=True) # 시트 이름으로 불러오기 load_ws = load_wb['Sheet1'] # URL을 읽어서 HTML를 받아오고, headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'} data = requests.get('https://www.genie.co.kr/chart/top200?ditc=D&ymd=20200403&hh=23&rtm=N&pg=1',headers=headers) # HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만듦 soup = BeautifulSoup(data.text, 'html.parser') # select를 이용해서, tr들을 불러오기 songs = soup.select('#body-content > div.newest-list > div > table > tbody > tr') # movies (tr들) 의 반복문을 돌리기 # i는 2부터. 즉, 2번째 행부터 채웁니다. i = 2 for song in songs: # td 태그 중 클래스가 number인 녀석의 텍스트 중 0, 1번째 글자를 가져오고 좌우 공백을 제거합니다. rank = song.select_one('td.number').text[0:2].strip() # td 태그 중 클래스가 info인 녀석 바로 아래, 클래스가 title, ellipsis 중복으로 붙은 a 태그를 가져옵니다. # 그리고 a 태그의 텍스트 전부를 가져오고 좌우 공백을 제거합니다. title = song.select_one('td.info > a.title.ellipsis').text.strip() # td 태그 중 클래스가 info인 녀석 바로 아래, 클래스가 artist, ellipsis 중복으로 붙은 a 태그를 가져옵니다. # 그리고 a 태그의 텍스트 전부를 가져옵니다. artist = song.select_one('td.info > a.artist.ellipsis').text # cell에 값들을 입력하기 load_ws.cell(i, 1, rank) load_ws.cell(i, 2, title) load_ws.cell(i, 3, artist) # i를 하나 키워서 다음 행에 적을
```

```
수 있게 합니다. i += 1 # 저장은 마지막에 한번만 하면 된다!
load_wb.save("mystock.xlsx")
```

Python

▼ 11) [미니프로젝트] 한달 간의 순위 변동을 그려보기

1. 10번 엑셀 저장하기를 30번 하면 되겠죠?

```
#미리 만들어놓은, dates dates = ['20200301', '20200302',
'20200303', '20200304', '20200305', '20200306', '20200307',
'20200308', '20200309', '20200310', '20200311', '20200312',
'20200313', '20200314', '20200315', '20200316', '20200317',
'20200318', '20200319', '20200320', '20200321', '20200322',
'20200323', '20200324', '20200325', '20200326', '20200327',
'20200328', '20200329', '20200330']
```

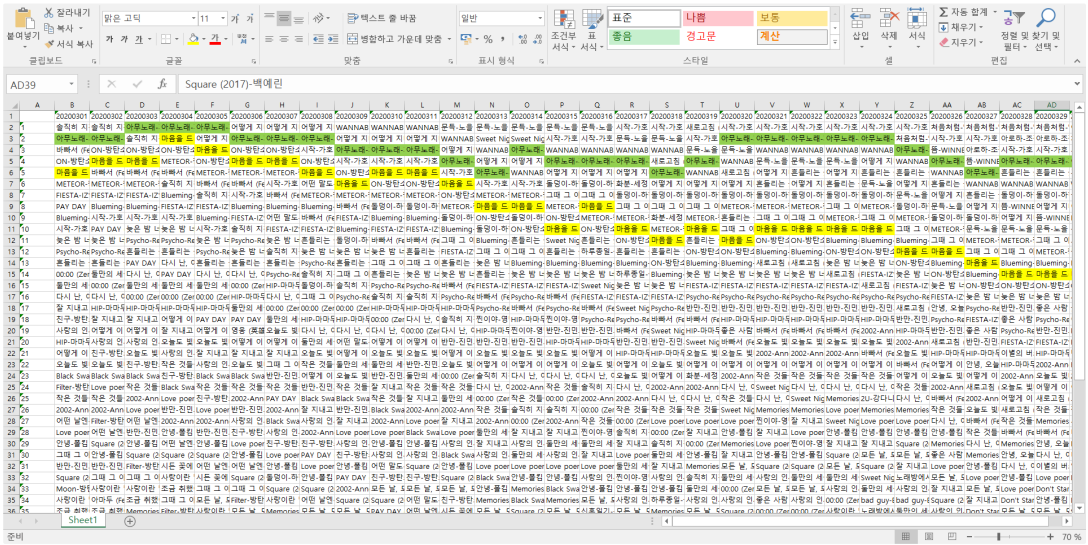
Python

```
import requests from bs4 import BeautifulSoup from openpyxl
import load_workbook # data_only=True로 해줘야 수식이 아닌 값으로 받
아온다. load_wb = load_workbook("mystock.xlsx", data_only=True)
# 시트 이름으로 불러오기 load_ws = load_wb['Sheet1'] dates =
['20200301', '20200302', '20200303', '20200304', '20200305',
'20200306', '20200307', '20200308', '20200309', '20200310',
'20200311', '20200312', '20200313', '20200314', '20200315',
'20200316', '20200317', '20200318', '20200319', '20200320',
'20200321', '20200322', '20200323', '20200324', '20200325',
'20200326', '20200327', '20200328', '20200329', '20200330'] j
= 2 for date in dates: # URL을 읽어서 HTML를 받아오고, headers = {
'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64;
x64)AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.86 Safari/537.36'} data =
requests.get('https://www.genie.co.kr/chart/top200?
ditc=D&ymd='+date+'&hh=23&rtm=N&pg=1', headers=headers) #
HTML을 BeautifulSoup이라는 라이브러리를 활용해 검색하기 용이한 상태로 만듦
soup = BeautifulSoup(data.text, 'html.parser') # select를 이용
해서, tr들을 불러오기 songs = soup.select('#body-content >
div.newest-list > div > table > tbody > tr') # 날짜 찍기
load_ws.cell(1, j, date) # movies (tr들) 의 반복문을 돌리기 i = 2
for song in songs: # td 태그 중 클래스가 number인 녀석의 텍스트 중 0,
1번째 글자를 가져오고 좌우 공백을 제거합니다. rank =
song.select_one('td.number').text[0:2].strip() # td 태그 중 클래
스가 info인 녀석 바로 아래, 클래스가 title, ellipsis 중복으로 붙은 a 태그
를 가져옵니다. # 그리고 a 태그의 텍스트 전부를 가져오고 좌우 공백을 제거합니다.
title = song.select_one('td.info >
a.title.ellipsis').text.strip() # td 태그 중 클래스가 info인 녀석
바로 아래, 클래스가 artist, ellipsis 중복으로 붙은 a 태그를 가져옵니다. #
그리고 a 태그의 텍스트 전부를 가져옵니다. artist =
song.select_one('td.info > a.artist.ellipsis').text
load_ws.cell(i, j, title+'-'+artist) i += 1 j += 1
load_wb.save("mystock.xlsx")
```

Python

2. 그러면, 이런 변동표를 그려볼 수 있어요~!

👉 지코의 '아무노래'와, 아이유의 '마음을 드려요'가 한달동안 어떻게 순위가 변동됐는지, 알 수 있어요!



[1시간] : 본 프로젝트 - 주기적으로 주식 정보를 가져와서 메일로 보내기

"자, 본 게임에 들어갑시다."

*참고! 실제 주식 트레이딩을 할 때에는, 0.01초가 중요하기에, 증권사 API를 사용한대요.
그치만 아래 방법으로는 항공권, 뉴스, 공연 등에도 다양하게 응용이 가능하답니다!

▼ 12) [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?

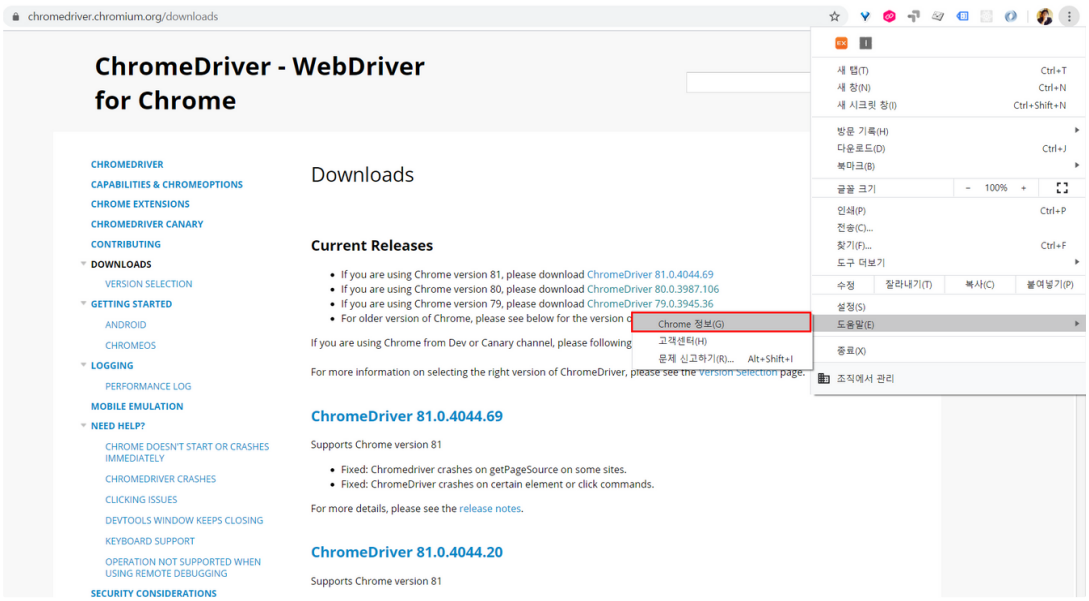
   - 아이디와 비밀번호를 자동으로 입력하고 클릭까지 하게 하려면?
   - 클릭으로 파일 다운로드까지 자동으로 하게 하려면?
   - 내가 원하는 시간에 영화 표 예매까지 자동으로 하게 하려면?

   이럴 때 필요한 녀석이 바로 Selenium(셀레니움)입니다!

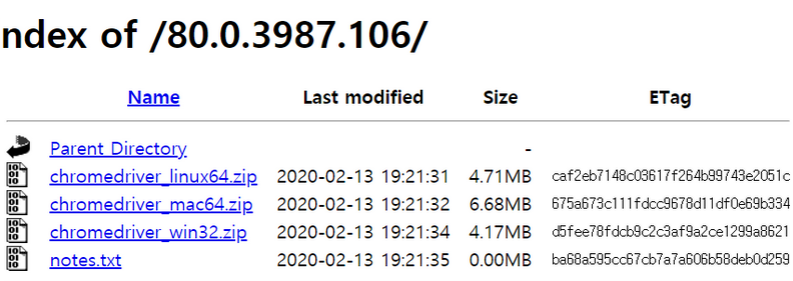
2. 브라우저 자동화를 위해선 'chromedriver'라는 것이 필요해요! (다운로드 링크)

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

   - 내 크롬 버전을 확인하기



   - 버전에 맞는 드라이버 다운로드 하기



3. 기본 코드 - 여기서 출발하세요!

```
설치해야 하는 라이브러리: selenium
```

Python

```
from selenium import webdriver from bs4 import BeautifulSoup
# 내장 라이브러리이므로 설치할 필요가 없습니다. 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) # 크롬을 종료합니다. driver.quit()
```

Python

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()
```

Python

▼ 13) [브라우저 자동화] 여러 주식 정보 가져오기 + 브라우저 안보이게

1. 12번을 여러번 돌리면 되겠죠?

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

Python

```
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 받아오
기까지 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()
```

Python

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)
```

Python

3. 완성코드

```python
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 받아오
기까지 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()
```

Python

▼ 14) [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!

```
설치해야 할 라이브러리: schedule
```

Python

```
import schedule def job(): print('여기에 할 일을 넣기') def run():
schedule.every(1).seconds.do(job) #1초에 한번씩 실행 (5분은 너무 길
어서..^^;) while True: schedule.run_pending() if __name__ ==
"__main__": run()
```

Python

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 =
'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) 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()
```

Python

▼ 15) [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 검색해 보세요! 거의 대부분 존재할 거에요 😎

2. 사전 작업

   ▶ G메일의 앱 비밀번호를 만들고, 받아오기

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 = "bumkyu.lee@gmail.com" # 내 비밀번호를 입력합니다. my_password
= "wpfzfpqiubxjnjun" # 이메일 받을 상대방의 주소를 입력합니다. you =
"bk.lee@spartacodingclub.kr" ## 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 내용
을 작성합니다. html = '이렇게 알림을 줄 수 있죠!' # 이메일 내용의 타입을 지
정합니다. 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()
```

Python

4. 완성코드: 15초 마다 돌면서 조건에 맞으면 메일로 알려주기

   👉 조건은, 전일 대비 4% 이상 오르면 - 으로 잡아뒀어요!

```
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 = "bumkyu.lee@gmail.com" # 내 비밀번호
를 입력합니다. my_password = "wpfzfpqiubxjnjun" # 이메일 받을 상대방의
주소를 입력합니다. you = "bk.lee@spartacodingclub.kr" ## 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+' 주식
을 한번 보세요!' # 이메일 내용의 타입을 지정합니다. part2 =
```

```
MIMEText(html, 'html') # 이메일 form에 작성 내용을 입력합니다
msg.attach(part2) ## 여기에서 코드 작성이 끝납니다. # Gmail을 통해 전달
할 것임을 표시합니다. s = smtplib.SMTP_SSL('smtp.gmail.com') # 계정
정보를 이용해 로그인합니다. s.login(me, my_password) # 이메일을 발송합니
다. 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 =
'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(15).seconds.do(job)
#10초에 한번씩 실행 while True: schedule.run_pending() if
__name__ == "__main__": run()
```

Python

[혼자해보기]

> 👻 **"아직 뭐가 뭔지 모르겠다고요? 하고 싶은 걸 상상하고, 만들어보세요!"**

- 구글링과 집념만 있으면, 못 할 게 없답니다.

- 그래도 아직 혼자서는 어렵다면?

  [질문도 하세요!] : 웹페이지 오른쪽 아래 톡버튼(링크)을 통해서!

  > 🤔 집에서 공부 좀 하려는데, 막히면 안되죠! 사회적 거리두기 종료까지(~04/19) 3명의 튜터가 붙어서 질문에 답 드리기로 하였습니다. 마음껏 물어봐주세요. 최대한 빠르게 답 드릴게요!