

## Chapter. 02

## 데이터 크롤링

# 고급주제

M T W T F S S

2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

**FASTCAMPUS**  
**ONLINE**

금융공학/퀀트 I

강사. 서찬웅

# I 이번 시간에 배울 내용

이번 시간에는 잠시 쉬어가는 시간을 갖도록 하겠습니다.

여기서 쉬어간다는 의미가 논다는 의미보다 앞으로 공부할 내용을 잠시 리뷰 하는 시간을 가지면서

준비하는 시간이 더 정확한 표현일지 모르겠습니다.

해당 구체적인 내용은 앞으로 계속 공부할 것입니다.

1. pandas는 무엇인가?
2. 웹 사이트 로그인(인증) 은 어떻게 하면 되고, 편하게 할 방법은 무엇인가?



# I DataFrame에 대해서 알아보겠습니다.

- 파이썬으로 데이터를 실제로 분석을 한다면 numpy 기반의 수치형 데이터 타입위에 DataFrame형식으로 분석을 할 가능성이 매우 큽니다. 사실 이 패키지를 사용할 수 있기 때문에 파이썬이 분석에 활용한 이유이기도 합니다.
- 파이썬으로 분석하기라는 말은 곧 DataFrame 형식의 데이터 타입으로 구성된 데이터를 전처리하고 의미있는 값을 찾기 위해서 기계학습(딥러닝), 통계 모형을 실행하여 예측 및 의미를 찾는 것입니다.
- DataFrame을 만든 개발자를 소개합니다.

## Short biography

Since 2007, I have been creating fast, easy-to-use data wrangling and statistical computing tools, mostly in the Python programming language. I am best known for creating the pandas project and writing the book *Python for Data Analysis*. I am also a contributor to the Arrow, Kudu (incubating), and Parquet projects within the Apache Software Foundation. I was the co-founder and CEO of DataPad. I later spent a couple years leading efforts to bring Python and Hadoop together at Cloudera. I'm now working for Two Sigma in New York.



## Open source projects

**pandas** (website): Python in-memory data wrangling, preparation, and analytics

- I created pandas and am its Benevolent Dictator for Life

# l pandas의 기능중 read\_html() 메소드를 알아보겠습니다.

- read\_html의 역할은 웹 페이지에서 table 형태로 되어 있는 데이터를 읽어서 바로 pandas의 데이터 형식인 DataFrame 형식으로 저장합니다. 저희가 BeautifulSoup이나 다른 작업을 할 필요가 없습니다.
- 네이버 주식에서 삼성전자의 시세를 가져오는 예제로 설명하겠습니다.

The image is a composite of three parts. On the left is a screenshot of the Samsung Electronics stock page on Naver, showing various financial metrics and a table of daily stock prices. In the center is a screenshot of a browser's developer tools network tab, showing a list of network requests. A red box highlights the 'Network' tab, and a red arrow points to the 'F12' key. On the right, there is a large red text overlay that reads: 'F12를 누르고 네트워크 탭을 선택해 주세요' and '로그 유지는 체크해 주세요'.

**시간별 시세**

체결시간	체결가	전일비	매도	매수	거래량	변동량
15:57	44,200	▲ 300	44,200	44,150	11,620,372	415
15:56	44,200	▲ 300	44,200	44,150	11,619,957	6
15:55	44,200	▲ 300	44,200	44,150	11,619,951	5
15:52	44,200	▲ 300	44,200	44,150	11,619,946	38
15:51	44,200	▲ 300	44,200	44,150	11,619,908	106
15:49	44,200	▲ 300	44,200	44,150	11,619,802	46
15:48	44,200	▲ 300	44,200	44,150	11,619,756	51
15:47	44,200	▲ 300	44,200	44,150	11,619,705	5
15:46	44,200	▲ 300	44,200	44,150	11,619,700	20
15:45	44,200	▲ 300	44,200	44,150	11,619,680	187

**일별 시세**

날짜	종가	전일비	시가	고가	저가	거래량
2019.06.07	44,200	▲ 300	43,600	44,350	43,450	11,620,372
2019.06.05	43,900	▲ 450	44,050	44,200	43,700	12,464,135
2019.06.04	43,450	▼ 350	43,400	43,700	43,000	9,913,497
2019.06.03	43,800	▲ 1,300	42,950	43,900	42,500	15,466,580
2019.05.31	42,500	▼ 50	42,600	42,800	42,150	11,365,999
2019.05.30	42,550	▲ 750	42,200	42,700	42,150	11,766,018
2019.05.29	41,800	▼ 750	41,850	42,100	41,300	14,930,618
2019.05.28	42,550	▼ 100	42,550	42,950	42,150	24,506,881
2019.05.27	42,650	▼ 50	42,500	43,000	42,350	8,066,669
2019.05.24	42,700	▼ 1,150	43,800	43,800	42,400	13,966,305

**인기검색목록**

순위	검색어	조회수	변동량
1	NAVER	111,000	▲ 500
2	삼성전자	44,200	▲ 300
3	셀트리온	196,500	▲ 500
4	한국전력	25,700	—
5	백광소재	5,400	↑ 1,245
6	SK하이닉스	65,400	▲ 300
7	세아제강	64,900	▲ 300
8	넷마블	119,000	▼ 5,000
9	삼성바이오로...	305,500	▼ 6,500
10	사조해표	8,380	—

**네트워크 탭**

상태	방식	도메인	파일	원인	유형	전송량	크기	시간	지연성	속도 제한 없음	HAR
200	GET	api.finance...	itemSummary.nhn?itemcode=005930&c...	script	json	460 B	196 B	59 ms	1.37분	2.73분	
200	GET	polling.fina...	realtime.nhn?query=SERVICE_ITEM.00593...	script	plain	2.64 KB	2.42 KB	59 ms			
200	GET	api.finance...	itemSummary.nhn?itemcode=005930&c...	script	json	460 B	196 B	91 ms			
200	GET	polling.fina...	realtime.nhn?query=SERVICE_ITEM.00593...	script	plain	2.64 KB	2.42 KB	59 ms			
200	GET	api.finance...	itemSummary.nhn?itemcode=005930&c...	script	json	460 B	196 B	70 ms			
200	GET	polling.fina...	realtime.nhn?query=SERVICE_ITEM.00593...	script	plain	2.64 KB	2.42 KB	89 ms			

아래와 같이 네이버 주식에서 일별시세의 페이지 번호를 클릭해보겠습니다.

No	Method	URL	Type	Size	Time
GET	finance.nav...	newstock2.css?20190529210744	stylesheet	css	21.79 KB
200 GET	finance.nav...	common.css?20190529210744	stylesheet	css	3.39 KB
GET	finance.nav...	newstock3.css?20190529210744	stylesheet	css	14.25 KB
GET	finance.nav...	world.css?20190529210744	stylesheet	css	5.88 KB
200 GET	finance.nav...	jindo.min.js?20190529210744	script	js	캐시됨
200 GET	finance.nav...	lcslog.js?20190529210744	script	js	캐시됨
200 GET	lcs.naver.com	m?u=https://finance.naver.com/item/sise...&page=1	img	gif	410 B
200 GET	finance.nav...	sise_day.nhn?code=005930&page=1	subdocument	html	2.24 KB
200 GET	finance.nav...	newstock.css?20190529210744	stylesheet	css	캐시됨
200 GET	finance.nav...	common.css?20190529210744	stylesheet	css	캐시됨
200 GET	finance.nav...	layout.css?20190529210744	stylesheet	css	캐시됨
200 GET		ico_arrow_wh.gif	200	gif	Other
200 GET		sp_gnb_v14.png	200	png	Other
200 GET		fxview?eu=EU10041051&calp...	200	text/pl...	pc.veta.core...
200 GET		sise_day.nhn?code=005930&p...	200	docu...	Other
200 GET		newstock.css?20190529210744	200	styles...	sise_day.nhn...
200 GET		common.css?20190529210744	200	styles...	sise_day.nhn...
200 GET		layout.css?20190529210744	200	styles...	sise_day.nhn...
200 GET		main.css?20190529210744	200	styles...	sise_day.nhn...
200 GET		newstock2.css?20190529210744	200	styles...	sise_day.nhn...
200 GET		newstock3.css?20190529210744	200	styles...	sise_day.nhn...
200 GET		world.css?20190529210744	200	styles...	sise_day.nhn...

일별 시세의 페이지 번호를 클릭하면 네트워크 탭에서 일별 시세의 정보를 얻어 오기 위한 페이지가 표시가 됩니다.

크롬이나 파이어폭스에서 쉽게 html 혹은 document로 되어 있는 정보를 확인할 수 있습니다.

마우스 오른쪽 버튼을 눌러 주소를 복사하면 해당 페이지의 url 주소가 복사가 됩니다.

# read\_html() 메소드를 사용하여 봅시다.

파이어폭스(크롬)을 사용하여 일별 시세에 관한 주소를 가져왔습니다.

이 주소를 활용해서 쉽게 데이터를 가져오겠습니다.

```
1 import pandas as pd
2
3 url = "https://finance.naver.com/item/sise_day.nhn?code=005930&page=1"
4 result = pd.read_html(url)
```

1 line : pandas 라이브러리를 import하여 사용하는데 as로 별명을 붙여서 pd라고 사용하겠습니다.

3 line : 전장에서 복사한 주소를 입력합니다.

4 line : pandas의 read\_html() 메소드에 url를 넘겨 데이터를 result에 저장합니다.

	날짜	종가	전일비	시가	고가	저가	거래량
0	NaN	NaN	NaN	NaN	NaN	NaN	
1	2019.06.07	44200.0	300.0	43600.0	44350.0	43450.0	11620372.0
2	2019.06.05	43900.0	450.0	44050.0	44200.0	43700.0	12464135.0
3	2019.06.04	43450.0	350.0	43400.0	43700.0	43000.0	9913497.0
4	2019.06.03	43800.0	1300.0	42950.0	43900.0	42500.0	15466580.0
5	2019.05.31	42500.0	50.0	42600.0	42800.0	42150.0	11365999.0
6	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	2019.05.30	42550.0	750.0	42200.0	42700.0	42150.0	11766018.0
10	2019.05.29	41800.0	750.0	41850.0	42100.0	41300.0	14930618.0
11	2019.05.28	42550.0	100.0	42550.0	42950.0	42150.0	24506881.0
12	2019.05.27	42650.0	50.0	42500.0	43000.0	42350.0	8066669.0
13	2019.05.24	42700.0	1150.0	43800.0	43800.0	42400.0	13966305.0
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN

0 1 2 3 4 5 6 7 8 9 10 11  
0 1 2 3 4 5 6 7 8 9 10 다음 맨뒤]

결과는 옆 화면처럼 나옵니다.

해당 데이터 타입은 list형식으로 총 2개의 원소를 가지고 있습니다.

첫번째 원소는 해당 테이블을 가져온 DataFrame 형식이며,

두번째 원소는 페이지 번호가 붙은 DataFrame형식입니다.

우리는 첫번째 원소의 데이터만 사용하겠습니다.

# I DataFrame의 기능을 잠시 사용하겠습니다.

아래는 첫번째 원소의 결과입니다. NaN는 사용할 수 없는 결측치 입니다.

DataFrame에서는 결측치를 쉽게 제거해주는 dropna()라는 메소드가 존재합니다.

```
result[0]
```

	날짜	종가	전일비	시가	고가	저가	거래량
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2019.06.07	44200.0	300.0	43600.0	44350.0	43450.0	11620372.0
2	2019.06.05	43900.0	450.0	44050.0	44200.0	43700.0	12464135.0
3	2019.06.04	43450.0	350.0	43400.0	43700.0	43000.0	9913497.0
4	2019.06.03	43800.0	1300.0	42950.0	43900.0	42500.0	15466580.0
5	2019.05.31	42500.0	50.0	42600.0	42800.0	42150.0	11365999.0
6	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	2019.05.30	42550.0	750.0	42200.0	42700.0	42150.0	11766018.0
10	2019.05.29	41800.0	750.0	41850.0	42100.0	41300.0	14930618.0
11	2019.05.28	42550.0	100.0	42550.0	42950.0	42150.0	24506881.0
12	2019.05.27	42650.0	50.0	42500.0	43000.0	42350.0	8066669.0
13	2019.05.24	42700.0	1150.0	43800.0	43800.0	42400.0	13966305.0
14	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
result[0].dropna()
```

	날짜	종가	전일비	시가	고가	저가	거래량
1	2019.06.07	44200.0	300.0	43600.0	44350.0	43450.0	11620372.0
2	2019.06.05	43900.0	450.0	44050.0	44200.0	43700.0	12464135.0
3	2019.06.04	43450.0	350.0	43400.0	43700.0	43000.0	9913497.0
4	2019.06.03	43800.0	1300.0	42950.0	43900.0	42500.0	15466580.0
5	2019.05.31	42500.0	50.0	42600.0	42800.0	42150.0	11365999.0
9	2019.05.30	42550.0	750.0	42200.0	42700.0	42150.0	11766018.0
10	2019.05.29	41800.0	750.0	41850.0	42100.0	41300.0	14930618.0
11	2019.05.28	42550.0	100.0	42550.0	42950.0	42150.0	24506881.0
12	2019.05.27	42650.0	50.0	42500.0	43000.0	42350.0	8066669.0
13	2019.05.24	42700.0	1150.0	43800.0	43800.0	42400.0	13966305.0

# I 파이썬으로 사이트 인증하여 로그인을 해봅시다.

파이썬에서 사이트에 로그인을 하기 위해서는 post 방식으로 접속하여 인증을 할 수 있습니다. 아래 예제는 기본적으로 사이트에 로그인 하기 위한 구조입니다. 대형 포털 사이트는 추가적인 보안으로 해당 방법으론 로그인이 되지 않지만 일반 커뮤니티 사이트는 로그인을 할 수 있습니다.

```
import requests

# 로그인 정보를 dict 형태로 전달합니다.
payload = { 'user_id' : '아이디',
            'password' : '패스워드' }

header = {
    #header..
    'Referer': 'https://www.ppomppu.co.kr/zboard/login.php',
    'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:63.0) Gecko/20100101 Firefox/63.0'
}

URL = "https://www.ppomppu.co.kr/zboard/login_check.php?/"

# with 구문으로 context manager를 실행합니다.
# with 구문이 종료되면 자동으로 세션도 종료됩니다.
with requests.Session() as s:
    rt = s.post(URL, data=payload, headers= header)

    r = s.get('http://www.ppomppu.co.kr/zboard/zboard.php?id=market', headers = header)

# 접속 정보 출력
print (r.text)
```

모든 사이트가 그런것은 아니지만 header의 값으로 크롤러인지 정상 로그인인지 확인하는 경우가 있다. 일반 브라우저에서 접속하는 것처럼 header를 작성

해당 사이트의 장터는 로그인하지 않으면 정보를 볼수가 없다. with 구문 Session안에서 로그인하여 해당 장터의 데이터를 가져올수 있음



# ! CSRF에 대해서 알아보겠습니다.

특정 사이트에서는 보안을 위해서 CSRF 코드를 html 안에 넣는 경우도 있습니다. CSRF의 값은 해당 페이지에 접속 할 때마다 변경이 되기 때문에 크롤링 코드에 이를 반영해야 합니다. 아래는 post방식으로 로그인 할 때 CSRF 값도 같이 받는 사이트 예제입니다.

```
import requests
from bs4 import BeautifulSoup
```

```
main_url = "https://www.clie.net/service/"
```

```
payload = { 'userId' : '아이디',
            'userPassword' : '패스워드' }
```

```
with requests.Session() as s:
```

```
    r = s.get(main_url)
```

```
    bs_data = BeautifulSoup(r.text, "html.parser")
```

```
    rt = bs_data.find("input", {"name": "_csrf"})['value']
```

```
    dict_rt = { **{'_csrf': rt}, **payload }
```

```
    post_rt = s.post('https://www.clie.net/service/login', data = dict_rt)
```

```
</form>
<div class="account_option">
  <a class="button_myarticle" href="/service/mypage/myArticle" title="나의글보기"><span class="fa fa-archive"></span> 나의글</a>
  <form class="form_logout" name="logout" action="/service/logout" method="POST">
    <input type="hidden" name="_csrf" value="1a57f8ec-dd80-4cde-9d9d-b02189ee9f64" />
    <input class="button_logout" type="submit" value="로그아웃" title="로그아웃" />
  </form>
</div>
```

\_csrf의 값을 post에 같이 전달을 해야한다.  
해당 값은 접속 할 때마다 매번 변경이 되기 때문에  
with 구문의 같은 세션으로 접속하면 동일한 csrf 값으로 접속할 수 있다.  
해당 값은 BeautifulSoup으로 값을 가져와서  
dict형태로 저장하고 payload dict하고 하나의 dict로 통합하였다.  
두 dict를 통합하는 방법엔 \*\*을 이용하였다.

# I 정리

이번시간에 배운 내용

앞으로 많이 사용하게 될 pandas 라이브러리

그중에서 dataframe 데이터 형식

read\_html() 메소드를 이용한 웹 데이터 파싱

post 방식의 인증 방법

예제 2개 사이트

# I 정리

- 이번시간에 배운 내용
  - 앞으로 많이 사용하게 될 pandas 라이브러리
  - 그중에서 dataframe 데이터 형식
  - read\_html() 메소드를 이용한 웹 데이터 파싱
- post 방식의 인증 방법
  - 예제 2개 사이트
  - header 정보
  - csrf 정보

# 감사합니다