데이터분석 with 파이썬

웹 스크래핑 VS 웹 크롤링

윤수연

목차

- 웹 크롤링 개요
- 웹 페이지 가져오기
- BeautifulSoup 시작하기
- 태그(Tag) 탐색하기
- 웹사이트 구조 분석
- csv, excel 저장

웹 스크래핑 vs 웹 크롤링

웹크롤링

- 웹 페이지에서 내가 원하는 부분만 추출
- 웹 페이지 링크를 따라가며 데이터 수집
- HTML (Hyper Text Markup Language) 이해
- xPath 이해
- 크롬 웹 브라우저 활용 (크롬 설치)

웹 스크래핑 vs 웹 크롤링

웹 크롤링 - 페이지 태그 확인

- 크롬에서 네이버 오픈 -> 특정 엘리먼트에서 마우스 우클릭 -> 검사 명령어 실행 (단축키 F12)
- 개발자도구에서 화면 왼쪽 상단 첫번째 아이콘 클릭 후 탐색
- 특정 엘리먼트에서 마우스 우클릭 -> Copy -> Copy XPath -> 주소창에 클릭하여 테스트
- 특정 엘리먼트에서 마우스 우클릭 -> Copy -> Copy full XPath -> 주소창에 클릭하여 테스트

웹 크롤링 : 웹페이지 가져오기

requests, beautifulsoup4 설치하기

명령문	pip list pip install requests pip install beautifulsoup4	# 라이브러리 리스트 확인 # 설치되 있으면 안해도 됨 # 설치되 있으면 안해도 됨

requests

- requests는 HTTP 요청 처리를 도와주는 모듈이다

beautifulsoup4

- beautifulsoup4는 HTML과 XML 파일에서 데이터를 읽어내준다.
- 크롤링에 필요하다고 볼 수 있다.

웹 크롤링 : 웹페이지 가져오기

requests

- HTML 문서에 담긴 내용을 가져오도록 request(요청) 한다.
- 예를 들어 당근마켓 인기 중고 매물을 가져오고 싶다면

명령문	import requests res = requests.get('https://www.daangn.com/hot_articles') print(webpage.text)

- 위의 코드를 실행하면 HTML 문서 전체를 긁어서 출력해준다.

웹 크롤링 : BeautifulSoup 시작하기

- **HTML 및 XML 문서를 탐색해서 원하는 부분만 쉽게 뽑아낼 수 있는 파이썬 라이브러리**
  - 요청한 HTML을 프린트 해보면 매우 길고 알아보기 복잡하다.
  - HTML 문서에서 필요한 부분만 뽑아내는 방법이 크롤링이다.

명령문	from bs4 import BeautifulSoup webpage = requests.get("https://www.daangn.com/hot_articles") soup = BeautifulSoup(webpage.content, "html.parser") print(soup)

  - 웹페이지를 요청한 뒤, 여기서 받아낸 문서를 .content로 지정한 후 BeautifulSoup를 통해 soup라는 객체로 저장하면 된다.
- **구문 분석 : Python 언어로 HTML 및 XML을 처리하기 위한 파서(parser)**
  - 파서는 'html.parser' 또는 'lxml', 'html5lib' 등의 옵션을 사용할 수 있음.

웹 크롤링 : BeautifulSoup 시작하기

- **타이틀 가져오기 :** 태그로 구성된 트리에서 title 태그만 출력

명령문	print(soup.head.title)
결과	<title>당근마켓 중고거래 | 당신 근처의 당근마켓</title>

- ** p 태그 출력 : 첫번째 p 태그를 찾아줌.**

명령문	print(soup.p)
결과	당근마켓 앱에서 따뜻한 거래를 직접 경험해보세요!

- **# 텍스트만 깔끔하게 출력**

명령문	print(soup.p.string)
결과	당근마켓 앱에서 따뜻한 거래를 직접 경험해보세요!

웹 크롤링 : BeautifulSoup 시작하기

원하는 태그의 내용 가져오기

- Find 를 사용하면 원하는 태그의 정보만 가져올 수 있다.

명령문	print(soup.head.find('meta',{'name':'description'}))
결과	<meta content="당근마켓에서 거래되는 인기 중고 매물을 소개합니다. 지금 당근마켓에서 거래되고 있는 다양한 매물을 구경해보세요." name="description"/>

meta 태그의 content 내용을 가져옴.

명령문	print(soup.head.find('meta',{'name':'description'}).get('content'))
결과	당근마켓에서 거래되는 인기 중고 매물을 소개합니다. 지금 당근마켓에서 거래되고 있는 다양한 매물을 구경해보세요.

웹 크롤링 : BeautifulSoup 시작하기

find_all을 통해 원하는 부분 가져오기

- 원하는 부분을 모두 가져올 수 있는 .find_all( ) 을 사용
- 모든 링크의 텍스트와 주소 가져오기
- a 태그로 둘러싸인 텍스트와 a 태그의 href 속성을 출력

명령문	for link in soup.find_all('a'): print(link.text.strip(), link.get('href'))

결과값을 리스트로 돌려주는 태그

명령문	print(soup.find_all("h2"))

find_all을 통해 원하는 부분 가져오기

- class가 card-title 인 태그만 출력하기

명령문	soup.find_all(attrs={'class' : 'card-title'})

텍스트만 읽어오기

위에서 `soup.select(".card-title")`이라는 걸 통해 class가 card-title인 것들만 리스트로 텍스틀를 뽑아보고 싶다면`.get_text()`를 사용하여 그 부분만 추출하면 된다.

명령문	for x in range(0,10): print(soup.select(".card-title")[x].get_text())
결과	앞에서 10개만 출력

웹 스크래핑 : 네이버 코스피 시각총액 순위

- 네이버에서 코스피 시가총액 순위 검색

명령문	import csv import pandas as pd import requests from bs4 import BeautifulSoup url = 'https://finance.naver.com/sise/sise_market_sum.nhn' res = requests.get(url) res.raise_for_status() soup = BeautifulSoup(res.text, 'lxml') print(soup)
결과	웹페이지 보여줌

웹 스크래핑 : 네이버 코스피 시각총액 순위

- 네이버에서 코스피 시가총액 순위 검색

명령문	import csv import pandas as pd import requests from bs4 import BeautifulSoup url = 'https://finance.naver.com/sise/sise_market_sum.nhn' tables = pd.read_html(url, encoding='euc-kr') tables
결과	테이블 보여줌

웹 스크래핑 : 네이버 코스피 시각총액 순위

[pandas] read_html () 함수

- 함수의 첫번째 매개변수로 URL 입력
- 해당 URL 페이지 내 <table> 태그로 이루어진 테이블을 파싱 함
- 여러 개의 테이블이 파싱 될 수 있으므로 list 타입으로 반환

명령문	url = 'https://finance.naver.com/sise/sise_market_sum.nhn' tables = pd.read_html(url, encoding='euc-kr')

read_html 함수의 사용 예시

명령문	type(tables)
결과	list

명령문	len(tables)
결과	3

read_html 함수 list 타입 반환 형식

웹 스크래핑 : 네이버 코스피 시각총액 순위

- pandas 에서 테이블 검색

명령문	len(tables) # 테이블 길이 (개수)
결과	3

명령문	Tables[1] # 테이블[인덱스]

	N	종목명	현재가	전일비	등락률	액면가	시가총액	상장주식수	외국인비율	거래량	PER
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1.0	삼성전자	59200.0	100.0	-0.17%	100.0	3534111.0	5969783.0	56.00	14945221.0	18.52
2	2.0	SK하이닉스	84500.0	800.0	+0.96%	5000.0	615162.0	728002.0	48.26	4088479.0	26.91
3	3.0	삼성바이오로직스	740000.0	18000.0	-2.37%	2500.0	489621.0	66165.0	10.26	95669.0	142.55
4	4.0	NAVER	292000.0	6000.0	-2.01%	100.0	479649.0	164263.0	55.07	651551.0	68.07

웹 스크래핑 : 네이버 코스피 시각총액 순위

Read_html() 함수 예시 결과

```
tables
```

```
[     0      1          2        3          4       5
0   거래량    매수호가    거래대금(백만)  시가총액(억)     영업이익(억)  PER(배)
1    시가    매도호가       전일거래량  자산총계(억)     영업이익증가율  ROE(%)
2    고가   매수총잔량        외국인비율  부채총계(억)     당기순이익(억)  ROA(%)
3    저가   매도총잔량   상장주식수(천주)    매출액(억)     주당순이익(원)  PBR(배)
4   NaN     NaN         NaN    매출액증가율   보통주배당금(원)  유보율(%)

        N      종목명        현재가       전일비   ...          거래량     PER    ROE  토론실
0     NaN      NaN         NaN       NaN   ...          NaN     NaN    NaN  NaN
1     1.0    삼성전자     58900.0     200.0   ...   10864514.0   18.43   8.69  NaN
2     2.0   SK하이닉스     83100.0     100.0   ...    1882521.0   26.46   4.25  NaN
3     3.0    NAVER   305000.0    6500.0   ...     636847.0   71.10  10.56  NaN
4     4.0     LG화학   677000.0   18000.0   ...     348043.0  116.70   1.84  NaN
..    ...       ...         ...       ...   ...          ...     ...    ...  ...
76   49.0      오리온   133500.0       0.0   ...      50260.0   19.92  14.24  NaN
77   50.0  포스코케미칼    84400.0     600.0   ...     114615.0   99.53  11.57  NaN
78    NaN      NaN         NaN       NaN   ...          NaN     NaN    NaN  NaN
79    NaN      NaN         NaN       NaN   ...          NaN     NaN    NaN  NaN
80    NaN      NaN         NaN       NaN   ...          NaN     NaN    NaN  NaN

[81 rows x 13 columns],
   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]
```

comma 기준으로**3**개의 테이블로구성되어있음

웹 스크래핑 : 네이버 코스피 시각총액 순위

필요한 테이블 후처리

- "NaN"으로 이루어진 불필요한 행(row) 제거 필요
- "토론실 " 열(column) 제거 필요

```
[9]  tables[1]
```

	N	종목명	현재가	전일비	등락률	액면가	시가총액	상장주식수	외국인비율	거래량	PER	ROE	토론실
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1.0	삼성전자	58900.0	200.0	+0.34%	100.0	3516202.0	5969783.0	55.86	10864514.0	18.43	8.69	NaN
2	2.0	SK하이닉스	83100.0	100.0	-0.12%	5000.0	604970.0	728002.0	48.87	1882521.0	26.46	4.25	NaN
3	3.0	NAVER	305000.0	6500.0	+2.18%	100.0	501003.0	164263.0	55.39	636847.0	71.10	10.56	NaN
4	4.0	LG화학	677000.0	18000.0	+2.73%	5000.0	477910.0	70592.0	37.39	348043.0	116.70	1.84	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
76	49.0	오리온	133500.0	0.0	0.00%	500.0	52781.0	39536.0	44.20	50260.0	19.92	14.24	NaN
77	50.0	포스코케미칼	84400.0	600.0	+0.72%	500.0	51474.0	60988.0	7.39	114615.0	99.53	11.57	NaN
78	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
79	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
80	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

81 rows × 13 columns

웹 스크래핑 : 네이버 코스피 시각총액 순위

[DataFrame] drop() 함수

- 매개변수로 입력한 인덱스를 제거하는 함수
- axis=1 이라는 매개변수를 명시 할 경우 인덱스가 아닌 열(column)을 제거

명령문	kospi_table=tables[1] kospi_table = kospi_table.drop('토론실', axis=1) kospi_table

	N	종목명	현재가	전일비	등락률	액면가	시가총액	상장주식수	외국인비율	거래량	PER	ROE
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1.0	삼성전자	58900.0	200.0	+0.34%	100.0	3516202.0	5969783.0	55.86	10864514.0	18.43	8.69
2	2.0	SK하이닉스	83100.0	100.0	-0.12%	5000.0	604970.0	728002.0	48.87	1882521.0	26.46	4.25
3	3.0	NAVER	305000.0	6500.0	+2.18%	100.0	501003.0	164263.0	55.39	636847.0	71.10	10.56
4	4.0	LG화학	677000.0	18000.0	+2.73%	5000.0	477910.0	70592.0	37.39	348043.0	116.70	1.84
...	...	...	...	...	...	...	...	...	...	...	...	...
76	49.0	오리온	133500.0	0.0	0.00%	500.0	52781.0	39536.0	44.20	50260.0	19.92	14.24
77	50.0	포스코케미칼	84400.0	600.0	+0.72%	500.0	51474.0	60988.0	7.39	114615.0	99.53	11.57
78	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
79	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
80	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

"토론실" 열**(column)**이 제거된 모습

웹 스크래핑 : 네이버 코스피 시각총액 순위

[DataFrame] dropna() 함수

- NaN 값이 존재하는 행(row)을 제거하는 함수
- subset=['컬럼명'] 을 매개변수를 명시 할 경우 해당 열(column)이 NaN인 행을 제거

명령문	kospi_table = kospi_table.dropna(subset=['N']) kospi_table

	N	종목명	현재가	전일비	등락률	액면가	시가총액	상장주식수	외국인비율	거래량	PER	ROE
1	1.0	삼성전자	58900.0	200.0	+0.34%	100.0	3516202.0	5969783.0	55.86	10864514.0	18.43	8.69
2	2.0	SK하이닉스	83100.0	100.0	-0.12%	5000.0	604970.0	728002.0	48.87	1882521.0	26.46	4.25
3	3.0	NAVER	305000.0	6500.0	+2.18%	100.0	501003.0	164263.0	55.39	636847.0	71.10	10.56
4	4.0	LG화학	677000.0	18000.0	+2.73%	5000.0	477910.0	70592.0	37.39	348043.0	116.70	1.84
5	5.0	삼성바이오로직스	712000.0	27000.0	+3.94%	2500.0	471095.0	66165.0	10.16	73673.0	137.16	4.77
9	6.0	삼성전자우	50800.0	100.0	+0.20%	100.0	418026.0	822887.0	86.73	1157758.0	15.89	NaN
10	7.0	현대차	182500.0	4500.0	-2.41%	5000.0	389944.0	213668.0	31.65	1497648.0	26.30	4.32
11	8.0	셀트리온	257500.0	3000.0	+1.18%	1000.0	347609.0	134994.0	21.14	382364.0	86.85	11.19
12	9.0	카카오	381000.0	10500.0	+2.83%	500.0	335540.0	88068.0	31.74	579815.0	-217.71	-5.81
13	10.0	삼성SDI	437500.0	1000.0	+0.23%	5000.0	300845.0	68765.0	41.74	119304.0	160.73	2.94
17	11.0	LG생활건강	1508000.0	9000.0	+0.60%	5000.0	235522.0	15618.0	45.39	12920.0	34.39	20.32
18	12.0	현대모비스	233500.0	2500.0	-1.06%	5000.0	221953.0	95055.0	43.74	226122.0	12.72	7.28

"N" 열**(column)**의 **NaN**행**(row)**을 제거한 모습

웹 스크래핑 : 네이버 코스피 시각총액 순위

- DataFrame] to_csv (), to_excel 함수
  - DataFrame 을 csv 또는 xlsx 파일로 저장하는 함수
  - 첫번째 매개변수로 csv, xlsx 파일이름 또는 저장경로 입력
  - index=False 를 명시할 경우 인덱스 값을 같이 저장 하지 않음

명령문	kospi_table.to_csv('kospi.csv', encoding='euc-kr', index=False) kospi_table.to_excel('kospi.xlsx', encoding='euc-kr', index=False)

웹 스크래핑 : 네이버 세계환율 조회

명령문	import pandas as pd url ='https://search.naver.com/search.naver?where=nexearch&sm=top_hty&fbm=1&ie=utf8&query=%ED%99%98%EC%9C%A8%EC%A1%B0%ED%9A%8C' tables = pd.read_html(url) tables

```
[    일    월    화    수    목    금    토
0 NaN NaN NaN NaN NaN NaN NaN,          통화명     매매기준율          전일대비       등락률
0       미국 USD  1163.50   전일대비보합0.00    0.00%
1   일본 JPY 100  1117.84   전일대비상승2.90   +0.26%
2     유럽연합 EUR  1371.59   전일대비하락6.34   -0.46%
3       중국 CNY   171.49   전일대비하락0.32   -0.19%
4       영국 GBP  1495.68  전일대비하락13.84   -0.92%
5       호주 AUD   845.22   전일대비하락3.20   -0.38%
6      캐나다 CAD   879.21   전일대비하락2.83   -0.32%
7     뉴질랜드 NZD   781.23   전일대비하락5.99   -0.76%]
```

웹 스크래핑 : 네이버 세계환율 조회

- pandas 에서 테이블 검색

명령문	len(tables) # 테이블 길이 (개수)
결과	2

명령문	Tables[1] # 테이블[인덱스]

	통화명	매매기준율	전일대비	등락률
0	미국 USD	1163.50	전일대비보합0.00	0.00%
1	일본 JPY 100	1117.84	전일대비상승2.90	+0.26%
2	유럽연합 EUR	1371.59	전일대비하락6.34	-0.46%
3	중국 CNY	171.49	전일대비하락0.32	-0.19%
4	영국 GBP	1495.68	전일대비하락13.84	-0.92%
5	호주 AUD	845.22	전일대비하락3.20	-0.38%
6	캐나다 CAD	879.21	전일대비하락2.83	-0.32%
7	뉴질랜드 NZD	781.23	전일대비하락5.99	-0.76%

웹 스크래핑 : 네이버 세계환율 조회

- pandas 에서 테이블 dataframe으로 저장

명령문	df=tables[1] df

	통화명	매매기준율	전일대비	등락률
0	미국 USD	1163.50	전일대비보합0.00	0.00%
1	일본 JPY 100	1117.84	전일대비상승2.90	+0.26%
2	유럽연합 EUR	1371.59	전일대비하락6.34	-0.46%
3	중국 CNY	171.49	전일대비하락0.32	-0.19%
4	영국 GBP	1495.68	전일대비하락13.84	-0.92%
5	호주 AUD	845.22	전일대비하락3.20	-0.38%
6	캐나다 CAD	879.21	전일대비하락2.83	-0.32%
7	뉴질랜드 NZD	781.23	전일대비하락5.99	-0.76%

웹 스크래핑 : 웹 스크래핑 후 CSV 파일로 저장하기

- csv 파일 저장

명령문	df.to_csv('exchange_rate.csv')

- csv 파일 읽어오기

명령문	x = pd.read_csv('exchange_rate.csv') x.head(10)

- Excel 파일 저장

명령문	df.to_excel('exchange_rate.xlsx')

웹 스크래핑 : Excel 파일로 저장하기

- Excel 파일 저장

명령문	df.to_excel('exchange rate.xlsx')

- Excel 파일 읽어오기

명령문	x = pd.read_excel('exchange rate.xlsx') x.head(10)

THANK YOU FOR YOUR ATTENTION