파이썬을 이용한 데이터수집 및 스마트공장 견학

Crawling의 기초
BeautifulSoup

2021년 1월 13일

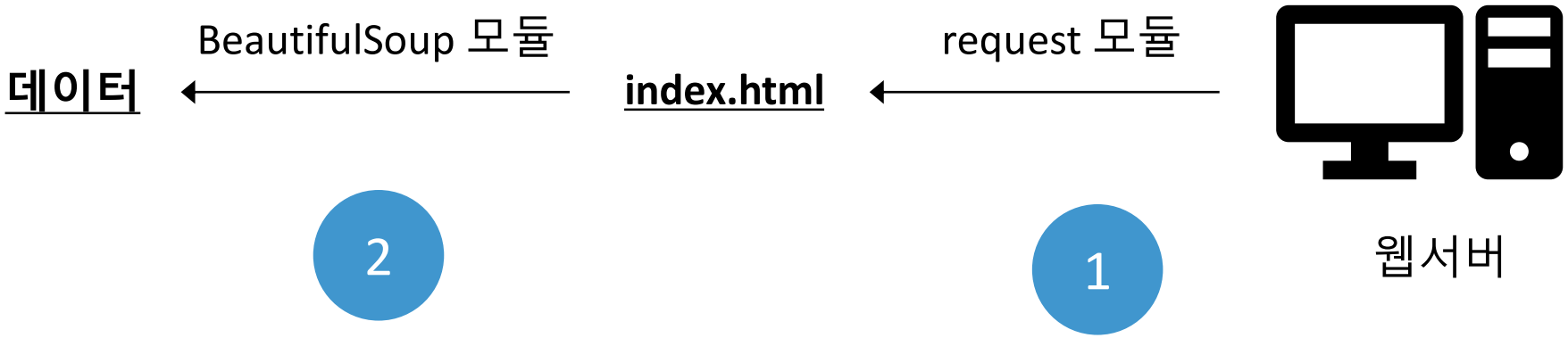
안재관

금일 목표

- Selector를 이용해 HTML 요소 골라내기
- BeautifulSoup에 selector를 사용해서 요소 추출
- for문을 사용해서 한 페이지에서 요소들을 반복 추출

HTML 소스 다운로드 및 분석

- **웹 스크래핑을 하려면 1) 웹 페이지로부터 HTML 파일을 받은 후 2) 파서를 사용하여 원하는 데이터가 있는 tag를 파싱해야 함**
  - 1) 웹 페이지 다운로드 → request 모듈
  - 2) 웹 페이지 파싱 → BeautifulSoup



HTML 소스 다운로드 및 분석

- **Python으로 웹페이지의 HTML 소스 받기**
  - requests 패키지를 사용

```
import requests
raw = requests.get("http://147.46.178.16:33333/table.html",
headers={'User-Agent': 'Mozilla/5.0'})
print(raw.text)
```

  - raw = requests.get("주소") : 주소를 방문하여 raw라는 변수에 소스를 저장
  - headers : 프로그램을 통해 방문하는게 아니라 브라우저라고 눈속임
  - 브라우저에서 소스 보기 했을때와 동일한 코드가 출력

```
<table border= "1 ">
  <tr>
    <th class="exampleClass1">Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
  <tr>
    <td>March</td>
```

```
<table border="1">
  <tr>
    <th class="exampleClass1">Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
  <tr>
    <td>March</td>
    <td>$180</td>
  </tr>
```

HTML 소스 다운로드 및 분석(예시: 네이버)

- **Python으로 웹페이지의 HTML 소스 받기**
  - requests 패키지를 사용

```
import requests
url=“http://www.naver.com”
response = requests.get(url)
print(type(response))
print(response.text)
```

```
1  import requests
2
3  url = "http://www.naver.com"    # 문자열
4  response = requests.get(url)    # requests 모듈의 get 함수 호출
5
6  print(type(response))
7  print(response.text)
8
```

```
<class 'requests.models.Response'>

<!doctype html>                              <html lang="ko" data-dark="false"> <head> <meta charset="utf-8"> <title>NAVER</title> <meta ht
tp-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=1190"> <meta name="apple-mobile-web-app-title" conte
nt="NAVER"/> <meta name="robots" content="index,nofollow"/> <meta name="description" content="네이버 메인에서 다양한 정보와 유용한 컨텐
츠를 만나 보세요"/> <meta property="og:title" content="네이버"> <meta property="og:url" content="https://www.naver.com/"> <meta propert
y="og:image" content="https://s.pstatic.net/static/www/mobile/edit/2016/0705/mobile_212852414260.png"> <meta property="og:description"
content="네이버 메인에서 다양한 정보와 유용한 컨텐츠를 만나 보세요"/> <meta name="twitter:card" content="summary"> <meta name="twitter:
title" content=""> <meta name="twitter:url" content="https://www.naver.com/"> <meta name="twitter:image" content="https://s.pstatic.ne
t/static/www/mobile/edit/2016/0705/mobile_212852414260.png"> <meta name="twitter:description" content="네이버 메인에서 다양한 정보와 유
용한 컨텐츠를 만나 보세요"/>  <link rel="stylesheet" href="https://pm.pstatic.net/dist/css/nmain.20220106.css"> <link rel="stylesheet"
href="https://ssl.pstatic.net/sstatic/search/pc/css/sp_autocomplete_210318.css"> <link rel="shortcut icon" type="image/x-icon" href="/f
```

BeatifulSoup 크롤링

- **Beautiful Soup**
  - Beautiful Soup is a Python library for pulling data out of HTML and XML files.
  - https://www.crummy.com/software/BeautifulSoup/bs4/doc/

- **Beautiful Soup vs. Request**
  - request 모듈은 http 프로토콜을 사용하여 웹 서버로부터 웹 페이지의 html 데이터를 파이썬 문자열 데이터 타입으로 변환해 줌
  - beautiful soup은 HTML로부터 데이터를 추출해 내는 라이브러리

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석**
  - bs4 패키지의 BeautifulSoup 모듈을 사용

```
from bs4 import BeautifulSoup
html = BeautifulSoup(raw.content, "html.parser", from_encoding="utf-8")
print(html)
```

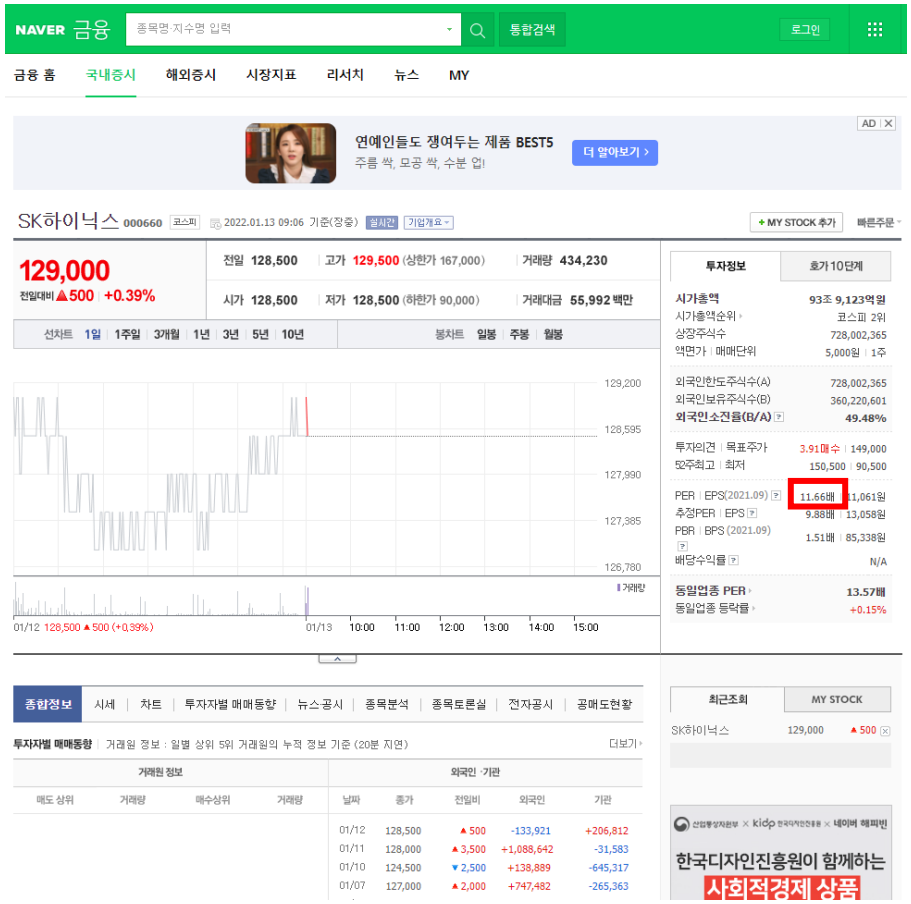
  - 출력 결과 자체는 동일하지만, html 변수 내에는 소스코드가 태그 단위로 분리되어 구조화되어 저장됨 → 이하 Soup변수라고 지칭
  - Soup변수에서는 추후 selector를 사용하여 원하는 정보를 빼오는 것이 가능함

- Soup변수.select("selector") : Selector에 해당하는 모든 태그들을 list로 추출
- Soup변수.select_one("selector") : Selector에 해당하는 첫 번째 태그를 추출
- Soup변수.text : 태그들을 제외하고 실제 텍스트만 추출

가나다라</br>이 저장되어 있으면 .text시 가나다라 가 추출됨
- Soup변수.get("속성명") : 해당 태그의 속성에 저장된 값을 추출
  네이버가 저장되어 있으면 .get("href")시 www.naver.com 이 추출됨

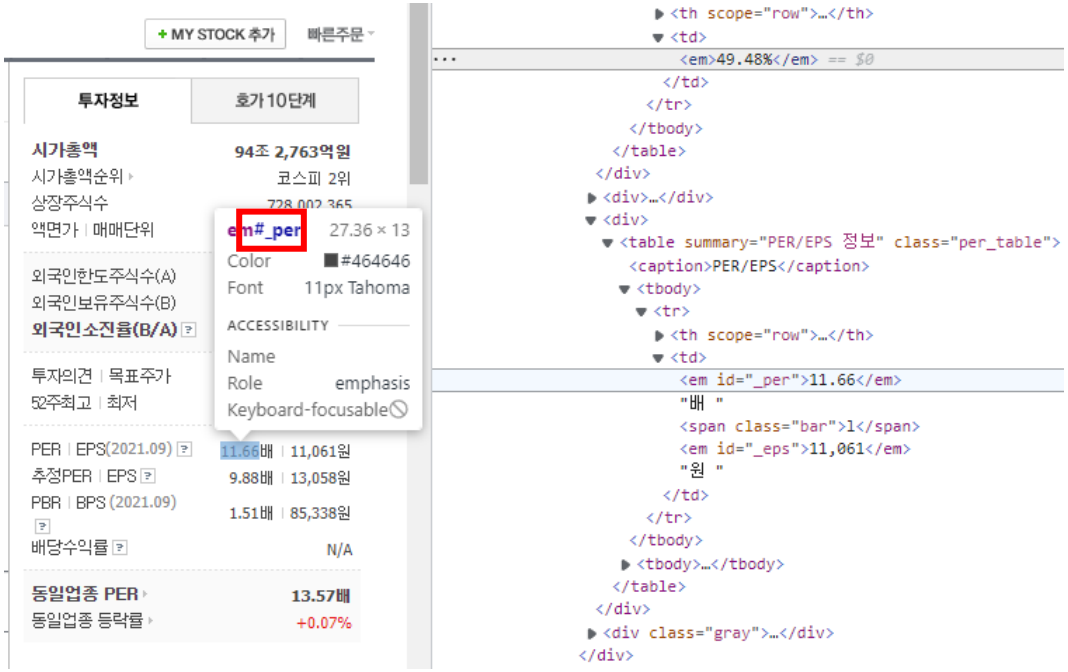
BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#0**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)
  - PER값 크롤링하기 실습



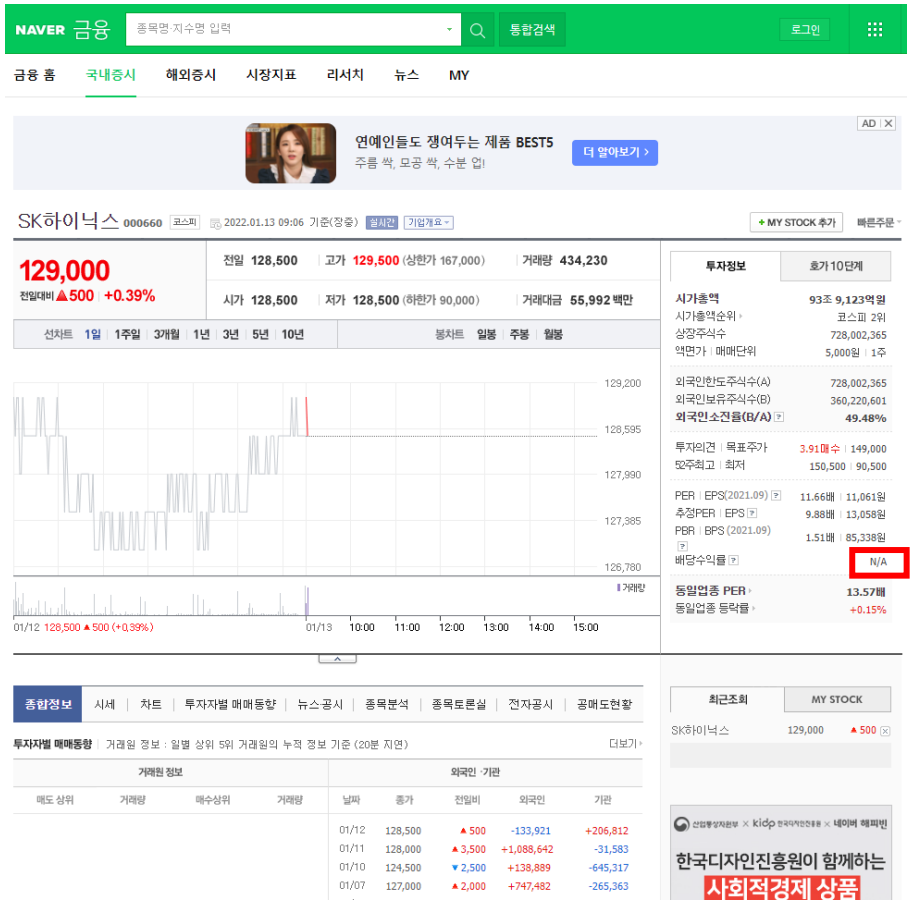
BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#0**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)
  - PER값 크롤링하기 실습



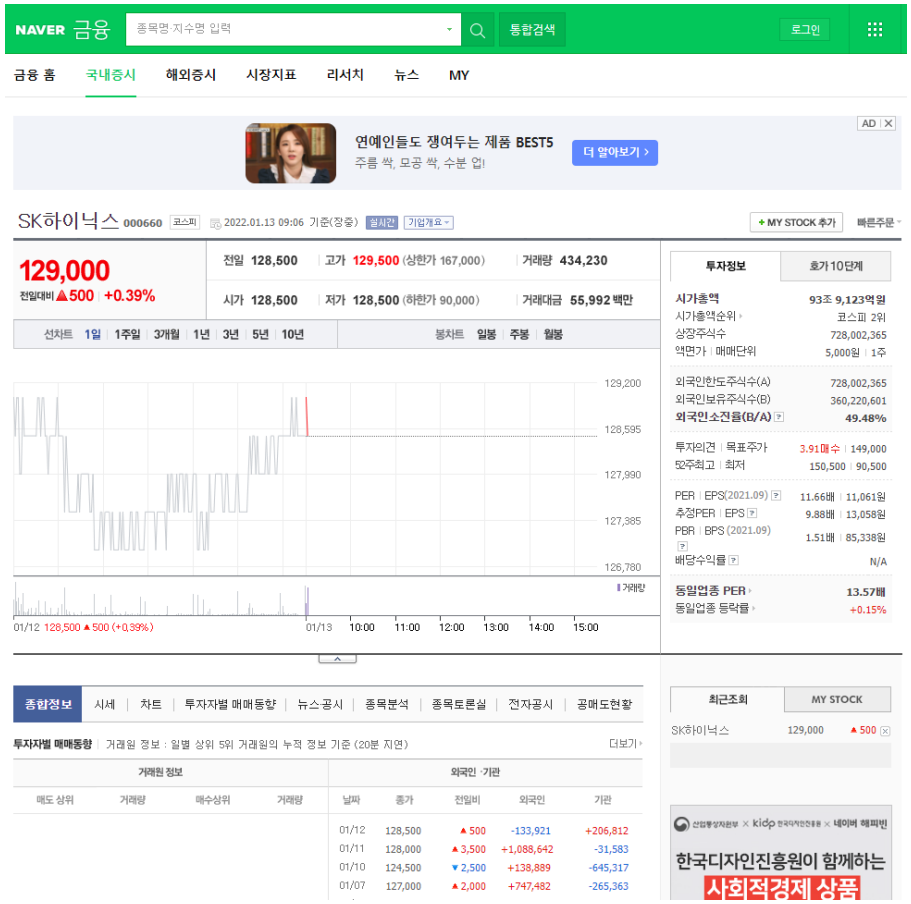
BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#0.5**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)
  - 배당률 크롤링하기 실습



BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#0.5**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)
  - 크롤링 함수 만들어보기



BeatifulSoup 크롤링: parser 관련

● Html.parser vs lxml vs xml vs html5lib

해석기	전형적 사용 방법	장점	단점

파이썬의 html.parser	`BeautifulSoup(markup, "html.parser")`
	
	`BeautifulSoup('<a></p>', 'html.parser')`
	하면 <a> 형태로 강제 변경되어 처리됨
		• 각종 기능 완비
		• 적절한 속도
		• 관대함 (파이썬3.2 이상)
			• 별로 관대하지 않음 (파이썬 3.2.2 이전 버전에서)

lxml의 HTML 해석기	`BeautifulSoup(markup, "lxml")`
	
	`BeautifulSoup('<a></p>', 'lxml')`
	하면 <html><body><a></body></html> 형태로 강제 변경되어 처리됨
		• 아주 빠름
		• 관대함
			• 외부 C라이브러리 의존

xml의 XML 해석기	`BeautifulSoup(markup, "xml")`
	
	`BeautifulSoup('<a><b />', 'xml')`
	하면
	<?xml version="1.0" encoding="utf-8" ?>
	<a><b/> 형태로 강제 변경되어 처리됨
		• 아주 빠름
		• 유일하게 XML 해석기 지원
			• 외부 C라이브러리 의존

html5lib	`BeautifulSoup(markup, html5lib)`
	
	`BeautifulSoup('<a></p>', 'html.parser')`
	하면
	<html><head></head><body><a> </body></html>
	형태로 강제 변경되어 처리됨
		• 아주 관대함
		• 웹 브라우저 방식으로 페이지를 해석함
		• 유효한 HTML5를 생성함
			• 아주 느림
			• 외부 파이썬 라이브러리 의존
			• 파이썬 2 전용

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#1**
  - http://www.kpu.ac.kr/contents/main/cor/kcollege.html 에서 한글 학과명의 태그들 전부 가져오기
    - 힌트 : 한글 학과명의 selector는 .label.ko 로 표현 가능했음

```
raw = requests.get("http://www.kpu.ac.kr/contents/main/cor/kcollege.html")
html = BeautifulSoup(raw.content, "html.parser", from_encoding="utf-8")
departments = html.select(".label.ko")
print(departments)
```

```
[<div class="label ko">기계공학과</div>, <div class="label ko">기계설계공학과</div>, <div
```

  - 태그를 제외하고 학과명들만 출력하기
    - 힌트 : Soup변수.select("selector")의 결과는 list임
      - 아래 두 코드의 결과는 동일함

```
for dept in departments:
    print(dept.text)
```

```
for dept in range(len(departments)):
    print(departments[dept].text)
```

```
기계공학과
기계설계공학과
메카트로닉스공학과
전자공학부
컴퓨터공학부
게임공학부
신소재공학과
생명화학공학과
디자인학부
경영학부
나노-광공학과
에너지·전기공학과
지식융합학부
```

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#2**
  - http://www.kpu.ac.kr/contents/main/cor/kcollege.html 에서 모든 학과의 정보를 출력, 각 행에는 하나의 학과 정보가 들어가며, 한글 학과명, 영문 학과명, 홈페이지 주소가 출력되어야 함
  - 힌트
    - 웹페이지에서 내가 뽑아야 할 정보가 (빨간 네모) 뭐뭐 있는지 보기

    - 공통적으로 뽑아야 할 정보들을 묶어주기 (초록 네모)
      -> 앞으로 "박스" 라고 표현
    - 박스들을 selector로 선택하고 for문 안에서 통해 빨간 네모들 추출
      - 구구단 예시로 비유
        - 기계공학과 초록 네모가 2단
        - 기계공학과 한글 학과명이 2 x 2, 영문 학과명이 2 x 3, 홈페이지 주소가 2 x 4

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#2**
  - http://www.kpu.ac.kr/contents/main/cor/kcollege.html 에서 모든 학과의 정보를 출력, 각 행에는 하나의 학과 정보가 들어가며, 한글 학과명, 영문 학과명, 홈페이지 주소가 출력되어야 함
  - 힌트
    - .select 함수로 학과 박스를 선택해서 departments 변수에 저장
    - 각 박스 내에서 추출할 정보를 하나의 변수로 설정
    - 한 줄에 예쁘게 출력하기 : print(변수1, "₩t", 변수2, "₩t", 변수3)
    - 한 학과의 출력이 완성되었으면 for문 안에 넣어서 모든 학과를 출력

```
departments = html.select("div.meta")
for dept in departments:
    koname = dept.select_one(".label.ko").text
    enname = dept.select_one(".label.en").text
    url = dept.select_one("a").get("href")
    print(koname, "\t", enname, "\t", url)
```

```
기계공학과    MECHANICAL ENGINEERING    http://subweb.kpu.ac.kr/machine/index.do
기계설계공학과    MECHANICAL DESIGN ENGINEERING    http://subweb.kpu.ac.kr/machineDe/index.do
메카트로닉스공학과    MECHATRONICS ENGINEERING    http://subweb.kpu.ac.kr/control/index.do
전자공학부    ELECTRONICS ENGINEERING    http://subweb.kpu.ac.kr/electronic/index.do
컴퓨터공학부    COMPUTER ENGINEERING    http://subweb.kpu.ac.kr/computer/index.do
```

BeatifulSoup 크롤링

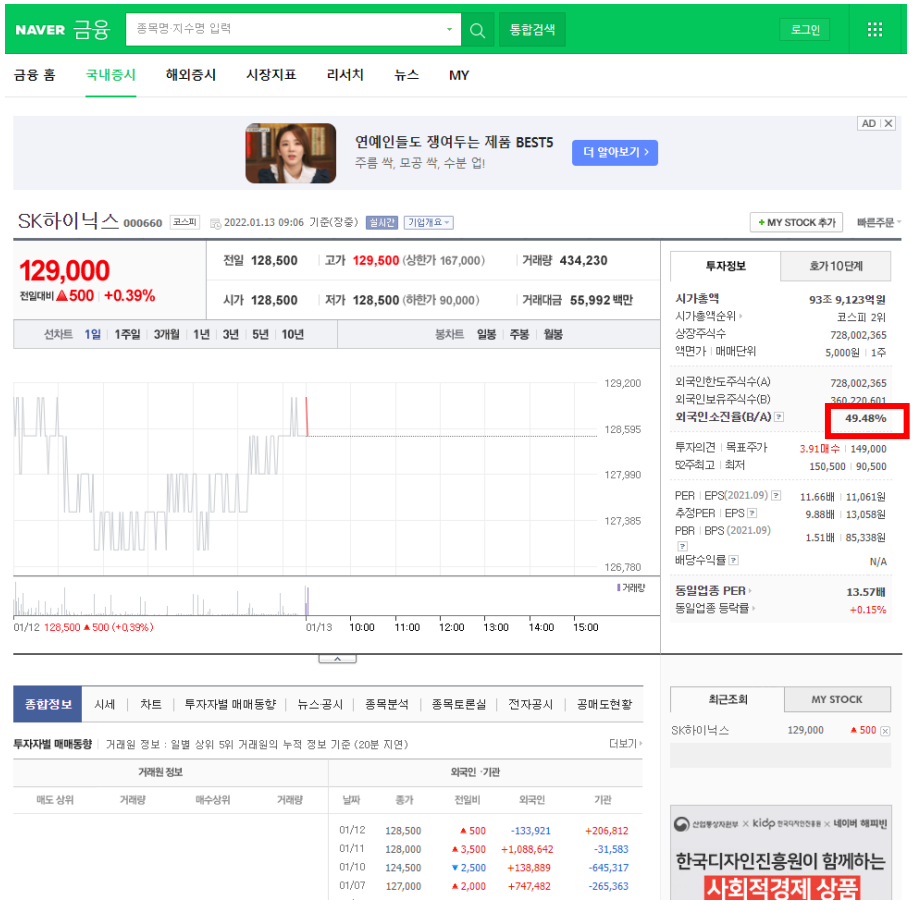
- **BeautifulSoup를 사용해서 소스 분석 실습#3**
  - https://sports.news.naver.com/index.nhn 에서 "오늘의 스포츠 NOW" 기사 6개의 정보 출력하기
  - 추출내용 : 스포츠 카테고리, 언론사, 기사제목

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#4**
  - http://market.cetizen.com/market.php?q=view&auc_no=24297415 세티즌 중고장터 (cetizen.com) 에서 "핸드폰 모델명" 추출하기

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#5**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)
  - ID가 없는 일반적인 경우는?



BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#5**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)
  - ID가 없는 일반적인 경우는?

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#5**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)
  - 복사한 CSS Selector를 붙여넣기

```
import requests
from bs4 import BeautifulSoup

url = "https://finance.naver.com/item/main.nhn?code=016360"
resp = requests.get(url)
html = resp.text

soup = BeautifulSoup(html, "html5lib")
tags = soup.select("#tab_con1 > div:nth-child(3) > table > tbody > tr.strong > td > em")
print(tags)
print(tags[0].text)
```

```
File "C:/Users/brayden/.spyder-py3/temp.py", line 9, in <module>
  tags = soup.select("#tab_con1 > div:nth-child(3) > table > tbody > tr.strong > td > em")

File "C:\Anaconda3\lib\site-packages\bs4\element.py", line 1609, in select
  for candidate in _use_candidate_generator(tag):

File "C:\Anaconda3\lib\site-packages\bs4\element.py", line 1570, in recursive_select
  for i in tag.select(next_token, recursive_candidate_generator):

File "C:\Anaconda3\lib\site-packages\bs4\element.py", line 1528, in select
  'Only the following pseudo-classes are implemented: nth-of-type.')

NotImplementedError: Only the following pseudo-classes are implemented: nth-of-type.
```

BeatifulSoup 크롤링

- **BeautifulSoup를 사용해서 소스 분석 실습#5**
  - http://finance.naver.com/item/main.nhn?code=000660(네이버금융:SK하이닉스)

  - CSS Selector 분석 후 수정하기
    - #tab_con1 > div:nth-child(3) > table > tbody > tr.strong > td > em (기존)
    - #tab_con1 > div:nth-of-type(2) > table > tbody > tr.strong > td > em (수정)