

Beautiful Soup

Auto Mobile Robot

Exported on 06/05/2024

Table of Contents

1	HTML 기초	4
1.1	언제나 시작은.....	4
1.2	파일 생성은 이제 쉽죠?	4
1.3	파일 이름은 test.html.....	4
1.4	HTML 을 작성해봅시다.	5
1.5	일단 HTML 이니까 웹브라우저로 볼까요?	5
1.6	오 이렇게 보이는 거였네요! HTML!!.....	5
1.7	HTML 구조	6
1.8	가벼운 실습.....	6
2	Beautiful Soup 설치.....	7
2.1	bs4.ipynb 파일 생성	7
2.2	가상환경 확인	7
2.3	Beautiful Soup 설치.....	7
2.4	Import BeautifulSoup.....	8
2.5	Troubleshooting	8
3	Beautiful Soup 기초.....	9
3.1	About Beautiful Soup.....	9
3.2	Beautiful Soup 으로 html 데이터 가져오기	10
3.2.1	test.html 과 비교.....	10
3.3	HTML Head	11
3.4	HTML Body.....	11
3.5	find()	11
3.6	find_all()	12
3.7	find() or find_all() with class	13
3.8	find() or find_all() with id	14
3.9	get_text()	14
3.10	string	15

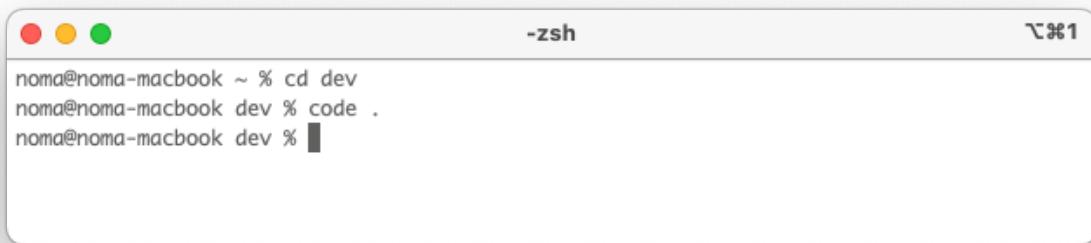
4	환율정보 가져오기	16
4.1	네이버 금융을 검색하고	16
4.2	시장지표 탭으로 이동	16
4.3	미국 달러 환율을 가져오고 싶은데..	17
4.4	일단 크롬 자랑	18
4.5	크롬 개발자 도구를 사용하면 된다!	18
4.6	오.. HTML 코드가 다보인다.....	19
4.7	그 값이 들어있는 태그를 찾아준다!	20
4.8	여기 있었네.. 미국 달러 환율값	20
4.9	URL 복사.....	21
4.10	HTML 가져오기~	21
4.11	아까 찾은 태그 정보를 활용해서~	22
5	위키백과 데이터 가져오기	23
5.1	구미호뎐 1938. 크흐~	23
5.2	구미호뎐 1938 위키백과 페이지로 이동	24
5.3	위키백과 페이지	24
5.4	URL 복사하기	25
5.5	URL 을 그대로 사용할 수 있어요	25
5.6	urllib.parse.quote()	26
5.7	가져오고 싶은 데이터는 특별 출연자들 이름!.....	26
5.8	일단은 무식하게	26
5.9	리스트에서 하나의 결과 가져오기	27
5.10	주요인물 데이터 확인	27
5.11	이쁘게 가져오기. List	28
6	고민해보기 - 네이버 영화순위	29

1 HTML 기초

일단 잘 모르지만, 파일을 작성하면서 살펴보겠습니다.

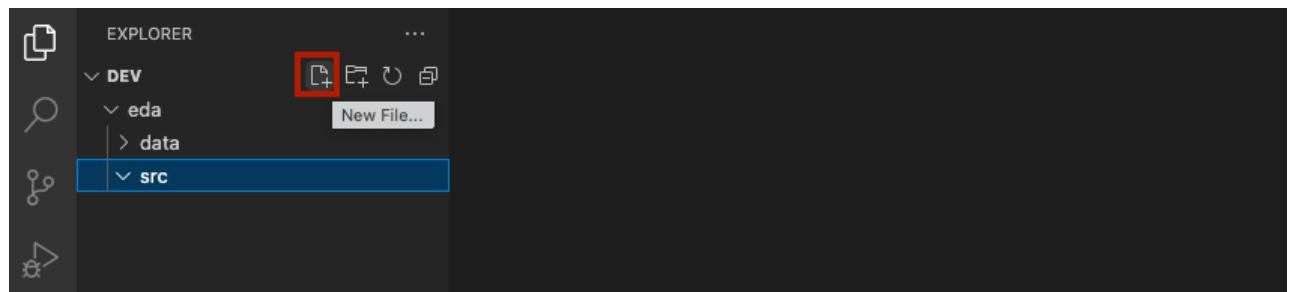
1.1 언제나 시작은..

터미널에서 dev로 이동 후, vscode 실행~

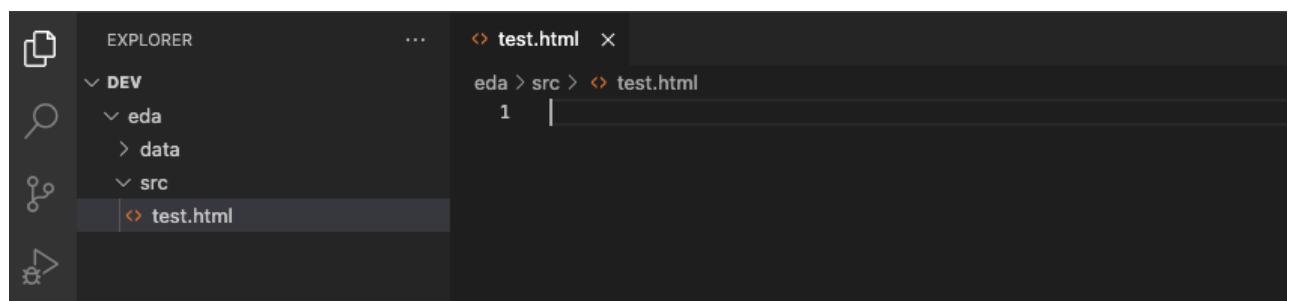


```
noma@noma-macbook ~ % cd dev
noma@noma-macbook dev % code .
noma@noma-macbook dev % [redacted]
```

1.2 파일 생성은 이제 쉽죠?

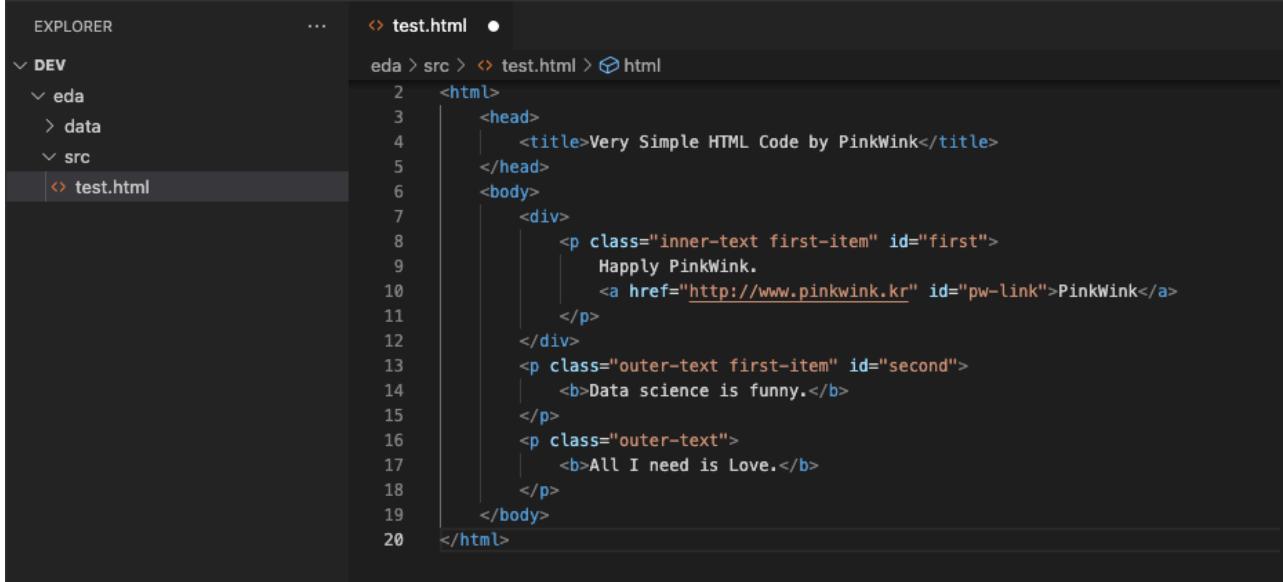


1.3 파일 이름은 test.html



1.4 HTML 을 작성해봅시다.

태그가 열리면 반드시 닫아야 합니다. 예 - <head> </head>

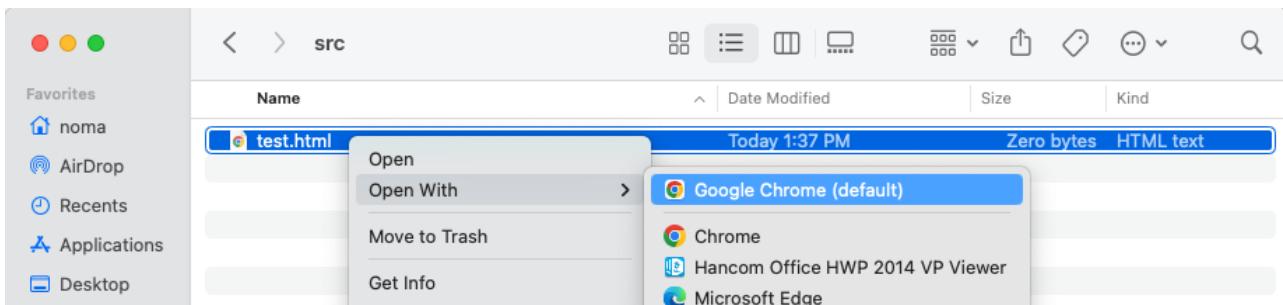


```

EXPLORER          ...  ◊ test.html ●
DEV
  eda
    data
  src
    test.html
eda > src > ◊ test.html > html
2   <html>
3     <head>
4       <title>Very Simple HTML Code by PinkWink</title>
5     </head>
6     <body>
7       <div>
8         <p class="inner-text first-item" id="first">
9           Happy PinkWink.
10          <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>
11        </p>
12      </div>
13      <p class="outer-text first-item" id="second">
14        <b>Data science is funny.</b>
15      </p>
16      <p class="outer-text">
17        <b>All I need is Love.</b>
18      </p>
19    </body>
20  </html>

```

1.5 일단 HTML 이니까 웹브라우저로 볼까요?



1.6 오 이렇게 보이는 거였네요! HTML!!

까만건 글자요.. 파란건 링크 구나..



1.7 HTML 구조

XML 이랑 언듯 비슷해보이지만, 조금 달라요~

```

<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>

```

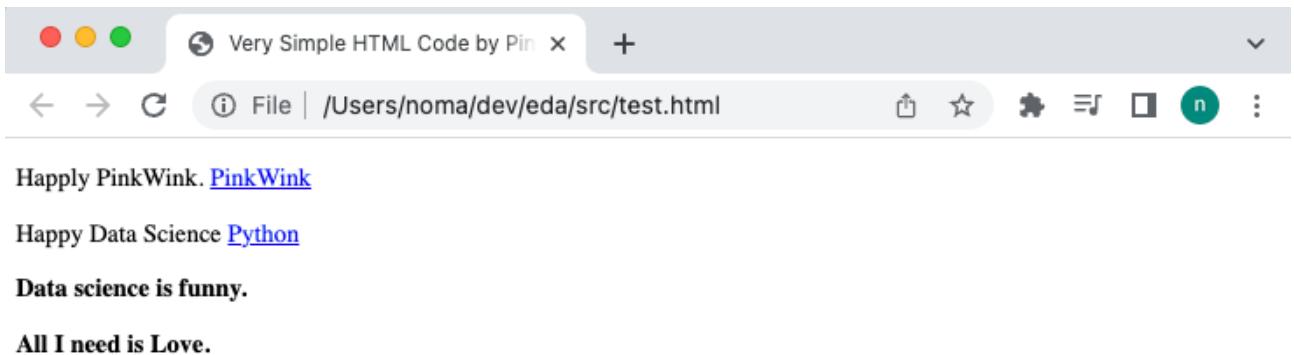
HTML 태그는 웹 페이지를 표현

HEAD 태그는 눈에 보이진 않지만 문서에 필요한 헤더 정보를 보관

BODY 태그에는 눈에 보이는 정보를 보관

1.8 가벼운 실습

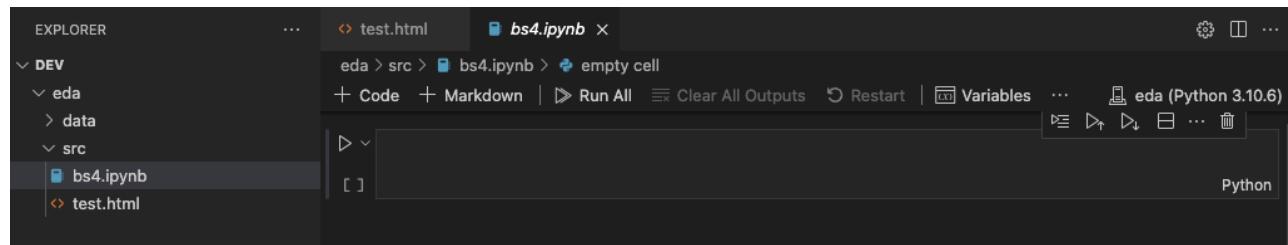
아래처럼 보이도록 HTML 을 수정해보세요. (파란건 파일 링크~ <http://www.python.org>)



2 Beautiful Soup 설치

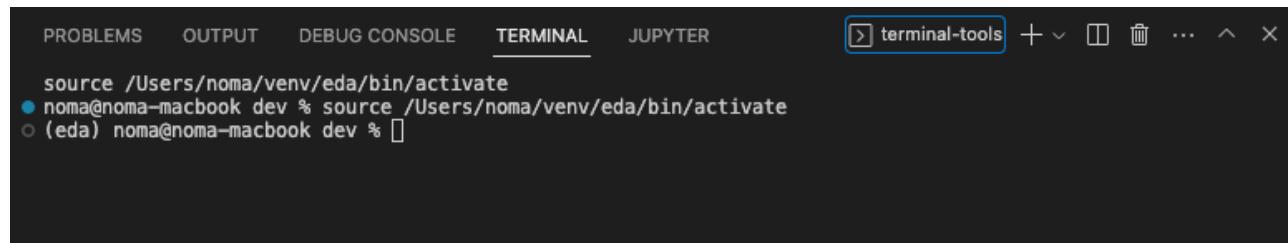
2.1 bs4.ipynb 파일 생성

가상환경은 이제 자연스럽게~ 사용가능?!



2.2 가상환경 확인

터미널을 하나 더 열면 새로운 가상환경이 활성화된 터미널이 열립니다.



2.3 BeautifulSoup 설치

```
pip install BeautifulSoup4
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER    terminal-tools + × ☰ ... ^ ×  
● noma@noma-macbook dev % source ~/venv/eda/bin/activate  
● (eda) noma@noma-macbook dev % pip install BeautifulSoup  
Collecting BeautifulSoup4  
  Using cached beautifulsoup4-4.12.2-py3-none-any.whl (142 kB)  
Collecting soupsieve>1.2  
  Using cached soupsieve-2.4.1-py3-none-any.whl (36 kB)  
Installing collected packages: soupsieve, BeautifulSoup4  
Successfully installed BeautifulSoup4-4.12.2 soupsieve-2.4.1  
  
[notice] A new release of pip available: 22.2.1 → 23.1.2  
[notice] To update, run: pip install --upgrade pip  
○ (eda) noma@noma-macbook dev %
```

2.4 Import BeautifulSoup

Beautiful Soup 모듈을 Import 해서 잘~ 설치가 되었는지 확인

The screenshot shows a Jupyter Notebook interface. The top bar displays the file names "test.html" and "bs4.ipynb". On the right, there are icons for settings, help, and more. Below the top bar, the sidebar shows the current path: "eda > src > bs4.ipynb > empty cell". The main menu includes "Code", "Markdown", "Run All", "Clear All Outputs", "Restart", "Variables", and "eda (Python 3.10.6)". In the code editor area, a cell has been run, showing the output: "from bs4 import BeautifulSoup". The output cell is labeled [1] and has a green checkmark indicating success, with a time of "0.0s". The language is identified as "Python".

```
from bs4 import BeautifulSoup
```

[1] ✓ 0.0s Python

2.5 Troubleshooting

만약 bs4 에러가 날 경우, 다음의 명령어로 설치해줍니다.

A screenshot of a terminal window from a code editor. The window has tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), JUPYTER, and a terminal tools tab. The terminal tools tab shows the command 'terminal-tools - noma'. Below the tabs, the terminal prompt shows '(eda) noma@noma-macbook ~ %' followed by the command 'python3 -m pip install bs4' which is partially typed and highlighted in blue.

3 Beautiful Soup 기초

3.1 About Beautiful Soup

HTML 혹은 XML 파일에서 데이터를 가져오기 위한 Python Library입니다.

출처: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

The screenshot shows a web browser window displaying the official Beautiful Soup documentation at <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. The title bar says "Beautiful Soup Documentation". The main content area is titled "Beautiful Soup Documentation". It contains several paragraphs of text explaining what Beautiful Soup is, how it works, and its features. A sidebar on the left lists various sections of the documentation, including "Table of Contents", "Beautiful Soup Documentation", "Getting help", "Quick Start", "Installing Beautiful Soup", "Installing a parser", "Making the soup", "Kinds of objects", "Tag", "NavigableString", "BeautifulSoup", "Special strings", "Comment", "For HTML documents", "Stylesheet", "Script", "Template", "For XML documents", and "Declaration". To the right of the text, there is a small cartoon illustration of two people in a kitchen setting, one holding a large umbrella.

3.2 BeautifulSoup 으로 html 데이터 가져오기

```

from bs4 import BeautifulSoup
[1]   ✓  0.0s          Python

page = open("test.html").read()
soup = BeautifulSoup(page, "html.parser")
print(soup.prettify())
[2]   ✓  0.0s          Python

...  <!DOCTYPE html>
<html>
<head>
<title>
    Very Simple HTML Code by PinkWink
</title>
</head>
<body>

```

- open.read() : 파일 읽기
- html.parser : BeautifulSoup의 html을 읽는 엔진 중 하나(lxml, html5lib도 많이 사용)
- prettify() : html 출력을 이쁘게 만들어 주는 기능

3.2.1 test.html 과 비교

<pre> 1 <!doctype html> 2 <html> 3 <head> 4 <title>Very Simple HTML Code by PinkWink</title> 5 </head> 6 <body> 7 <div> 8 <p class="inner-text first-item" id="first"> 9 Happy PinkWink. 10 PinkWink 11 <p class="inner-text second-item"> 12 Happy Data Science. 13 Python 14 </p> 15 </div> 16 <p class="outer-text first-item" id="second"> 17 18 Data Science is funny. 19 20 </p> 21 <p class="outer-text"> 22 23 All I need is Love. 24 25 </p> 26 </body> 27 </html> </pre>	<pre> <!DOCTYPE doctype html> <html> <head> <title> Very Simple HTML Code by PinkWink </title> </head> <body> <div> <p class="inner-text first-item" id="first"> Happy PinkWink. PinkWink </p> <p class="inner-text second-item"> Happy Data Science. Python </p> </div> <p class="outer-text first-item" id="second"> Data Science is funny. </p> <p class="outer-text"> All I need is Love. </p> </body> </html> </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.3 HTML Head

```
soup.head  
[4] ... <head>  
    <title>Very Simple HTML Code by PinkWink</title>  
  </head>  
Python
```

3.4 HTML Body

```
soup.body  
[3] ... <body>  
    <div>  
        <p class="inner-text first-item" id="first">  
            Happily PinkWink.  
            <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>  
        </p>  
        <p class="inner-text second-item" id="second">  
            Happy Data Science  
            <a href="http://python.org" id="python_link">Python</a>  
        </p>  
    </div>  
    <p class="outer-text first-item" id="second">  
        <b>Data science is funny.</b>  
    </p>  
    <p class="outer-text">  
        <b>All I need is Love.</b>  
    </p>  
  </body>  
Python
```

3.5 find()

조건에 맞는 태그 하나만 찾아주는 함수입니다.

```
[5]     soup.find("p")
Python
...
<p class="inner-text first-item" id="first">
    Happy PinkWink.
    <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>
</p>

1   <!doctype html>
2   <html>
3   |   <head>
4   |       <title>Very Simple HTML Code by PinkWink</title>
5   |   </head>
6   |   <body>
7   |       <div>
8   |           <p class="inner-text first-item" id="first">
9           |               Happy PinkWink.
10          |               <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>
11         </p>
```

3.6 find_all()

조건에 맞는 태그 모두를 찾아주는 함수입니다.

```
[6]     soup.find_all("p")
Python
...
[<p class="inner-text first-item" id="first">
    Happy PinkWink.
    <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>
</p>,
<p class="inner-text second-item" id="second">
    Happy Data Science
    <a href="http://python.org" id="python_link">Python</a>
</p>,
<p class="outer-text first-item" id="second">
<b>Data science is funny.</b>
</p>,
<p class="outer-text">
<b>All I need is Love.</b>
</p>]
```

```

6   <body>
7     <div>
8       <p class="inner-text first-item" id="first">
9         Happy PinkWink.
10        <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>
11      </p>
12      <p class="inner-text second-item">
13        Happy Data Science.
14        <a href="https://www.python.org" id="py-link">Python</a>
15      </p>
16    </div>
17    <p class="outer-text first-item" id="second">
18      <b>
19        Data Science is funny.
20      </b>
21    </p>
22    <p class="outer-text">
23      <b>
24        All I need is Love.
25      </b>
26    </p>

```

3.7 find() or find_all() with class

class 속성값으로 태그를 찾을 수 있습니다. (주의. class 는 python 예약어 이므로 class_ 로 표기)

```

[7]   soup.find_all(class_="outer-text")
          Python

...  [<p class="outer-text first-item" id="second">
      <b>Data science is funny.</b>
      </p>,
      <p class="outer-text">
      <b>All I need is Love.</b>
      </p>]

12    <p class="inner-text second-item" id="second">
13      Happy Data Science
14      <a href="http://python.org" id="python_link">Python</a>
15    </p>
16  </div>
17  <p class="outer-text first-item" id="second">
18    <b>Data science is funny.</b>
19  </p>
20  <p class="outer-text">
21    <b>All I need is Love.</b>
22  </p>
23  </body>
24  </html>

```

3.8 find() or find_all() with id

id 속성값으로 태그를 찾을 수 있습니다.

아이디는 고유값이기 때문에 결과 값이 하나인 경우가 대부분입니다. 이 경우 find_all() 함수를 사용할 이유가 없지만, List 타입으로 결과를 받고 싶은 경우 find_all() 을 사용하기도 합니다.

```
[8]     soup.find_all(id="first")                                         Python
...
[<p class="inner-text first-item" id="first">
    Happly PinkWink.
    <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>
</p>]
6     <body>
7         <div>
8             <p class="inner-text first-item" id="first">
9                 Happly PinkWink.
10                <a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>
11            </p>
12            <p class="inner-text second-item" id="second">
```

3.9 get_text()

태그 사이의 문자열을 가지고 옵니다.

```
[9]   for tag in soup.find_all("p"):
        print("-----")
        print(tag.get_text())
...
-----
Happy PinkWink.
PinkWink
-----
Happy Data Science
Python
-----
Data science is funny.
-----
All I need is Love.
```

3.10 string

태그 사이에 자식 태그가 존재하는 경우에도, 모든 문자열을 가지고 옵니다.

```
[10] link_list = soup.find_all("a")
link_list
...
[<a href="http://www.pinkwink.kr" id="pw-link">PinkWink</a>,
 <a href="http://python.org" id="python_link">Python</a>]

[11] for each in link_list:
        href = each["href"]
        text = each.string
        print(text + " -> " + href)
...
PinkWink -> http://www.pinkwink.kr
Python -> http://python.org
```

4 환율정보 가져오기

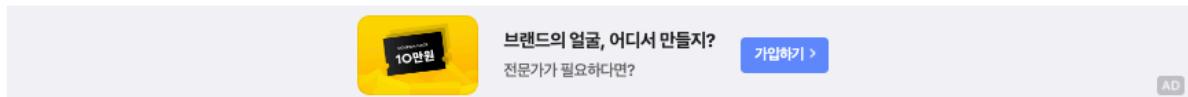
간단한 예제로 BeautifulSoup 과 좀더 친해져봅시다. feat. 크롬 개발자 도구

4.1 네이버 금융을 검색하고

The screenshot shows the Naver Finance homepage (finance.naver.com). At the top, there is a search bar with the placeholder "네이버 금융". Below the search bar is a navigation menu with links: 통합 (Integration), VIEW, 이미지 (Image), 지식iN (Knowledge iN), 동영상 (Video), 쇼핑 (Shopping), 뉴스 (News), 어학사전 (Dictionary), 지도 (Map), 책 (Books), and three dots for more options. The main content area features a banner for "네이버 금융" with sub-links for 시장지표 (Market Index), 뉴스 (News), 펀드 (Fund), 국내증시 (Domestic Stock Market), 리서치 (Research), and 해외증시 (Overseas Stock Market). Below the banner is a section titled "증권정보" (Securities Information) with dropdown menus for 주가지수 (Stock Index) and 주요증시 (Major Stock Markets).

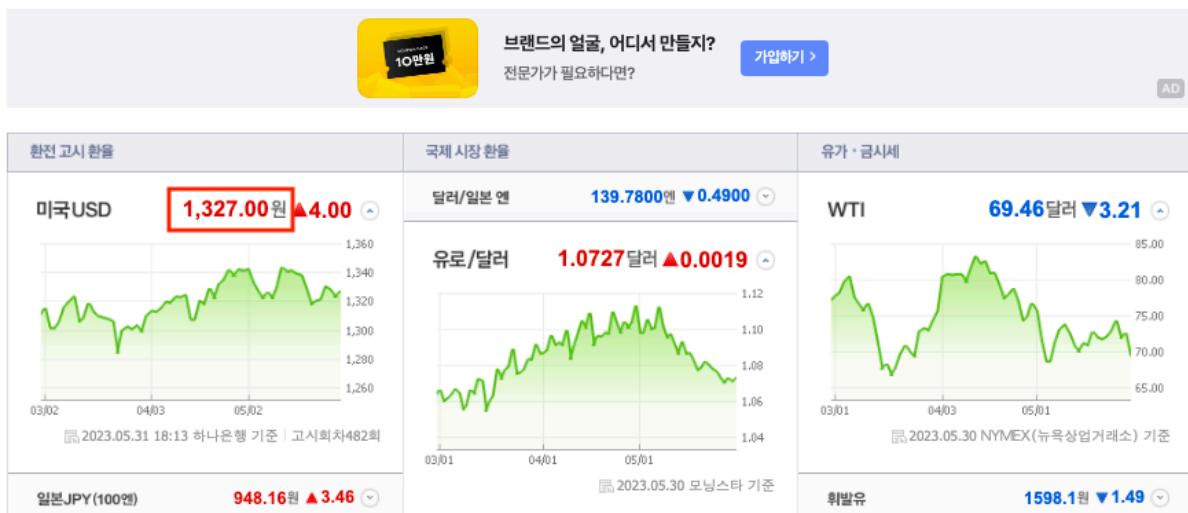
4.2 시장지표 탭으로 이동

- <https://finance.naver.com/marketindex/>



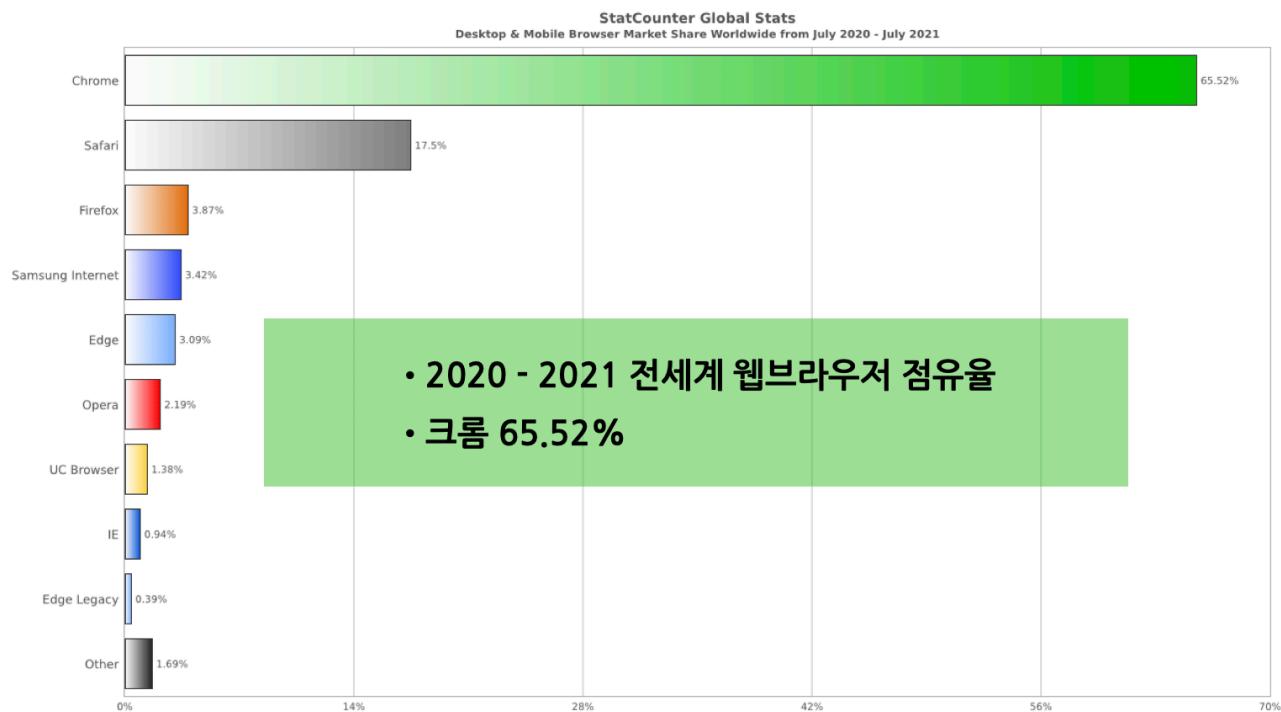
4.3 미국 달러 환율을 가져오고 싶은데..

파이썬 코드로 가져오고 싶지만 HTML 을 잘 모르겠다...



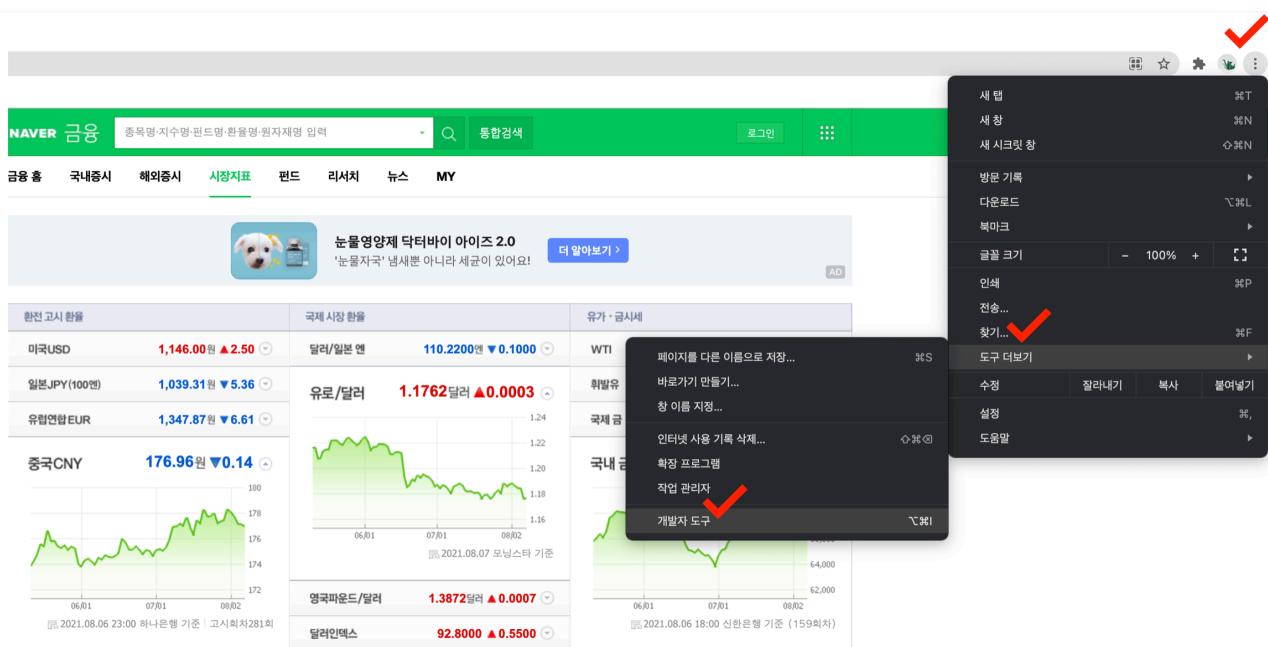
4.4 일단 크롬 자랑

출처: <https://gs.statcounter.com/browser-market-share/desktop-mobile/worldwide/#monthly-202007-202107-bar>



4.5 크롬 개발자 도구를 사용하면 된다!

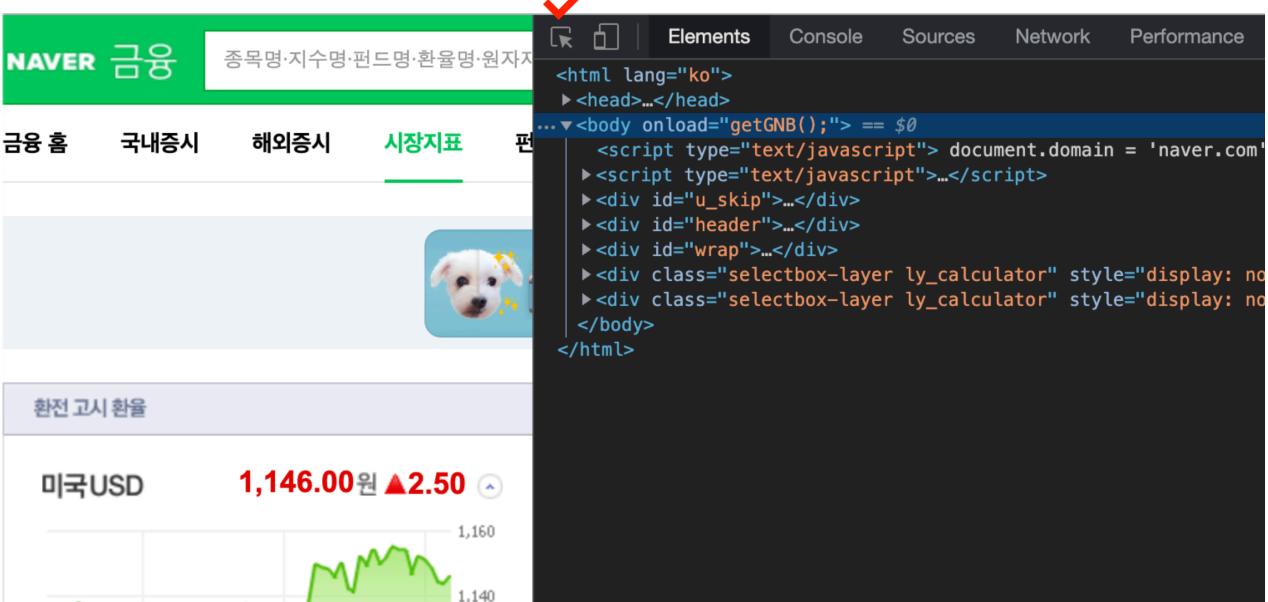
크롬 설정 > 도구 더보기 > 개발자 도구 선택



4.6 오.. HTML 코드가 다보인다.

아이콘을 선택하고 찾고 싶은 값을 웹화면에서 선택하면...

이 아이콘 선택



4.7 그 값이 들어있는 태그를 찾아준다!

The screenshot shows the NAVER Finance website for currency exchange rates. A red arrow points from the text "필요한 데이터 부분을 선택" (Select the required data part) to the current exchange rate value "1,146.00 원 ▲2.50". Below this, a green line chart tracks the USD exchange rate over time. To the right, the browser's developer tools (Elements tab) are open, displaying the HTML structure. A red box highlights the specific span element containing the value "1,146.00".

```

<html lang="ko">
  <head>...</head>
  <body onload="getGNB();">
    <script type="text/javascript"> document.domain = 'naver.com' </script>
    <script type="text/javascript" src="https://ssl.pstatic.net" ></script>
    <script type="text/javascript" src="https://ssl.pstatic.net" ></script>
    <div id="u_skip">...</div>
    <div id="header">...</div>
    <div id="wrap">
      <div class="banner_smart">...</div>
      <script type="text/javascript" src="https://ssl.pstatic.net" ></script>
      <script type="text/javascript" src="https://ssl.pstatic.net" ></script>
    </div>
    <div id="container" style="padding-bottom:0px;">
      <div class="market_include">
        <div class="market_data">
          <div class="market1">
            <div class="title">...</div>
            <!-- data -->
            <div class="data">
              <ul class="data_lst" id="exchangeList">
                <li class="on">
                  <a href="/marketindex/exchangeDetail.nhn?marketCd=USD&currCd=KRW" class="h_lst">...</a>
                  <div class="head_info_point_up">
                    <span class="value">1,146.00</span> == $0
                    <span class="txt_krw">...</span>
                    <span class="change">2.50</span>
                    <span class="blind">상승</span>
                  </div>
                <li>
                  ...
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

4.8 여기 있었네.. 미국 달러 환율값

The screenshot shows the same NAVER Finance website as before, but with a different focus. A red box highlights the span element with the class "value", which contains the value "1,146.00". To the right, the browser's developer tools show the full HTML structure of the page, with a red arrow pointing to the same span element.

```

<div class="title">...</div>
<!-- data -->
<div class="data">
  <ul class="data_lst" id="exchangeList">
    <li class="on">
      <a href="/marketindex/exchangeDetail.nhn?marketCd=USD&currCd=KRW" class="h_lst">...</a>
      <div class="head_info_point_up">
        <span class="value">1,146.00</span> == $0
        <span class="txt_krw">...</span>
        <span class="change">2.50</span>
        <span class="blind">상승</span>
      </div>
    <li>
      ...
    </li>
  </ul>
</div>

```

4.9 URL 복사

이제 코드로 처리하기 위해서 URL 을 복사

The screenshot shows a web browser window with the following details:

- Address Bar:** finance.naver.com/marketindex/
- Toolbar:** Includes icons for back, forward, search, and other navigation.
- Header:** NAVER 금융 (NAVER Finance) with a search bar for '종목명·지수명·펀드명·환율명·원자재명 입력' (Input stock name, index name, fund name, exchange rate name, commodity name).
- Menu:** Includes links for 앱 (App), YouTube, and various categories like 교육 (Education), 취업 (Job), 업무 (Business), and 퀴즈/과제 제작 (Quiz/Assignment Creation).
- Navigation:** tabs for 금융 홈 (Finance Home), 국내증시 (Domestic Stocks), 해외증시 (Overseas Stocks), 시장지표 (Market Indicators) (highlighted in green), 펀드 (Funds), 리서치 (Research), 뉴스 (News), and M.
- Content Area:**
 - A large banner for '눈물영양제 닥터바이' (Dr. Eye) with the text "'눈물자국' 냄새뿐 아니라" (Not just the smell of tear marks).
 - Exchange rate information: 환전 고시 환율 (Exchange rate) and 국제 시장 환율 (International market exchange rate).
 - Specific rates shown: 미국 USD 1,146.00원 ▲2.50 (USD 1,146.00 won, up 2.50) and 달러/일본 엔 110.220 (USD/JPY 110.220).

4.10 HTML 가져오기~

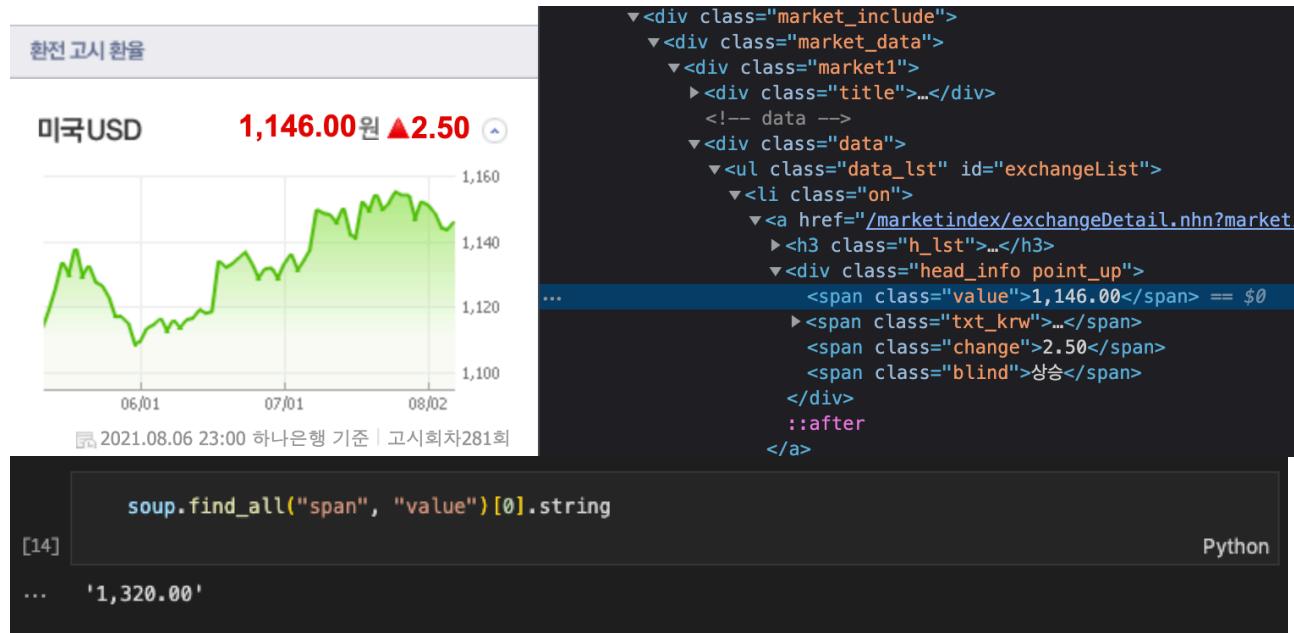
urllib 를 사용하여 해당 URL 의 HTML 데이터를 읽어와 봅시다.

```
[12] from bs4 import BeautifulSoup
      from urllib.request import urlopen
[13]
url = "https://finance.naver.com/marketindex/"
page = urlopen(url)

soup = BeautifulSoup(page, "html.parser")
print(soup.prettify())
...
<script language="javascript" src="/template/head_js.naver?referer=info.finance.naver.com&menu=</script>
<script src="https://ssl.pstatic.net/imgstock/static.pc/20230519195543/js/info/jindo.min.ns.1.5.3.e</script>
<script src="https://ssl.pstatic.net/imgstock/static.pc/20230519195543/js/jindo.1.5.3.element-text-</script>
<div id="container" style="padding-bottom:0px;">
```

4.11 아까 찾은 태그 정보를 활용해서~

`find_all()` 함수에서 태그 이름과 class 속성값을 사용하여 미국 달러 환율정보를 가져옵니다.



```
soup.find_all("span", "value")[0].string
```

```
[14]
```

```
Python
```

```
... '1,320.00'
```

5 위키백과 데이터 가져오기

한번 더 예제를 통해 BeautifulSoup 과 친해져 봅시다.

5.1 구미호뎐 1938. 크흐~



5.2 구미호뎐 1938 위키백과 페이지로 이동

The screenshot shows a Google search result for '구미호뎐 1938'. The top bar has the Google logo and a search input field containing '구미호뎐 1938'. Below the search bar are several links, with the first one being a Wikipedia page titled '구미호뎐 1938 - 위키백과, 우리 모두의 백과사전'. The page summary indicates it's a 2023 TV drama on tvN, directed by Kang Shin-hyo and Jo Nam-Young, produced by Studio Ondemand, and starring Lee Dong-uk, Kim Beom, Kim So-eun, and Ryu Kyung-soo.

5.3 위키백과 페이지

The screenshot shows the '구미호뎐 1938' Wikipedia page. The page title is '구미호뎐 1938' with a '15' rating icon. The page content includes sections for '기획의도' (Concept), '제작진' (Production Team), '제작사' (Production Company), '연출' (Directors), and '출연' (Actors). To the right of the main content, there is a sidebar with detailed information about the drama, such as its genre (Fantasy Action Romance), broadcast details (Korean, tvN, May 6, 2023 – June 11, 2023, 9:20 AM – 10:30 AM), and cast members (Park Jin-hyung, Son Chae-sung, Lee Dong-uk, Kim Beom, Kim So-eun, Ryu Kyung-soo).

5.4 URL 복사하기

구미호뎐 1938 위키백과 페이지의 URL 을 복사하여 셀에 붙여보면..

뭔가 이상하게 바뀌어서 나타나는데, 웹주소는 UTF-8 로 인코딩 되기 때문입니다.

구미호뎐 1938

목차 [숨기기]

문서 토론

처음 위치

위키백과, 우리 모두의 백과사전.

기획의도

제작진

구미호뎐 1938은 2023년 5월 6일부터 2023년 6월 11일까지 방영중인 tvN 토일드라마다.

https://ko.wikipedia.org/wiki/%EA%B5%AC%EB%AF%B8%ED%98%BB%EB%8E%90_1938

Python

5.5 URL 을 그대로 사용할 수 있어요.

그런데 코드 가독성이 몹시 떨어집니다.

```
[1] from bs4 import BeautifulSoup
from urllib.request import urlopen
[1] ✓ 0.0s Python
```



```
[2] url = "https://ko.wikipedia.org/wiki/%EA%B5%AC%EB%AF%B8%ED%98%BB%EB%8E%90_1938"
page = urlopen(url)

soup = BeautifulSoup(page, "html.parser")
print(soup.prettify())
[2] ✓ 0.6s Python
```



```
... <!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-
<head>
<meta charset="utf-8">
<title>
구미호뎐 1938 – 위키백과, 우리 모두의 백과사전
```

5.6 urllib.parse.quote()

urllib.pars.quote() 는 글자를 URL 형식으로 인코딩 해 줍니다. 이제 좀 알아보겠습니다.

```

import urllib
from urllib.request import Request

url = "https://ko.wikipedia.org/wiki/{search_words}"
req = Request(url.format(search_words = urllib.parse.quote("구미호뎐 1938")))
res = urlopen(req)

soup = BeautifulSoup(res, "html.parser")
print(soup.prettify())

```

[3] ✓ 1.0s Python

... <!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main">
<head>
<meta charset="utf-8"/>
<title>
구미호뎐 1938 – 위키백과, 우리 모두의 백과사전
</title>

5.7 가져오고 싶은 데이터는 특별 출연자들 이름!

크롬 개발자 도구를 사용하여 특별 출연 데이터 위치를 확인합니다.

특별 출연 [편집]

- 김영훈 : 삼천갑자 동방삭 역 (†) (1회)
- 조보아 : 남지야 / 이아음 역
- 황석정 : 최승자 역
- 안재모 : 애인시대 김두한 역
- 김법래 : 득각귀 역

시청률 [편집]

최저 시청률과 최고 시청률은 시청률 조사회사와 지역별로 시청률에 차이가 있을 수 있습니다.

... == \$0

 ::marker
 김영훈
 " " : 삼천갑자 동방삭 역 (†) (1회)

 ::marker
 조보아
 " " : 남지야 / 이아음 역"

5.8 일단은 무식하게

'ul' 태그 하위에 특별 출연자들 이름이 있는 것을 확인했으니 find_all() 을 이용하여 가져와봅니다. 너무 많네요.

```
[7]   for idx, each in enumerate(soup.find_all("ul")):
        print("[ " + str(idx) + " ] " + "=====")
        print(each.prettify())
[7]   ✓ 0.0s                                         Python
...
[0] =====
<ul class="vector-menu-content-list">
<li class="mw-list-item" id="n-mainpage-description">
<a accesskey="z" href="/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:%EB%8C%80%EB%AC%B8" title="대문"
<span>
    대문
</span>
</a>
</li>
<li class="mw-list-item" id="n-recentchanges">
```

5.9 리스트에서 하나의 결과 가져오기

`find_all()`로 가져온 리스트는 아래와 같이 하나씩 가져올 수 있습니다.

```
[8]   soup.find_all("ul") [10]
[8]   ✓ 0.0s                                         Python
...
<ul class="vector-toc-list" id="toc-제작자-sublist">
</ul>
```

5.10 주요인물 데이터 확인

60번째 결과가 특별출연자들 정보입니다.

```
[51]   soup.find_all("ul") [60]
[51]   ✓ 0.0s                                         Python
...
<ul><li><a href="/wiki/%EC%98%81%ED%9B%88_(1997%EB%85%84)" title="영훈 (1997년)">영훈 (1997년)</a>
<li><a href="/wiki/%ED%99%A9%EC%84%9D%EC%A0%95" title="황석정">황석정</a> : 최승자 역
<li><a href="/wiki/%EC%95%88%EC%9E%AC%EB%AA%A8" title="안재모">안재모</a> : 야인시대
<li><a href="/wiki/%EA%B9%80%EB%B2%95%EB%9E%98" title="김법래">김법래</a> : 독각귀 역
<li><a href="/wiki/%EC%A1%B0%EB%B3%B4%EC%95%84" title="조보아">조보아</a> : 남지아
```

5.11 이쁘게 가져오기. List

```
list = []

ul = soup.find_all("ul")[60]
for a in ul.find_all("a"):
    list.append(a.string)

list
[54] ✓ 0.0s
```

The screenshot shows a Jupyter Notebook cell with the following Python code:

```
list = []

ul = soup.find_all("ul")[60]
for a in ul.find_all("a"):
    list.append(a.string)

list
```

The cell has a status bar indicating [54] and a green checkmark with 0.0s. To the right of the code, it says "Python". The output of the code is shown below the cell:

```
['김영훈', '황석정', '안재모', '김법래', '조보아']
```

6 고민해보기 - 네이버 영화순위

네이버에서 평점순으로 영화 순위 정보를 가져오는 방법을 고민해봅시다.