



외부연동 데이터 수집 (웹 크롤링 활용 데이터 수집)

김 효 관 |

외부연동 데이터 수집

단원 개요

[단원명]

외부연동 데이터 수집

[단원 소개]

- 데이터 분석 시 회사 내부에서 접근할 수 있는 데이터 (파일, 데이터베이스) 만으로는 한계가 있다. 외부에서 활용가능한 포털 내 데이터 나 공개한 데이터를 수집하는 방법을 익히고 다양한 각도로 분석 모델을 만들어가는 방법을 학습한다.

[교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
웹 크롤링 활용 데이터 수집	<ul style="list-style-type: none">- 외부 데이터 수집의 필요성을 이해한다.- 웹 포털 내 존재하는 데이터 수집 방법을 실습한다.- 브라우저 자동화 라이브러리를 활용한 데이터 수집 자동화 방법을 실습한다.
공공데이터 포털 데이터 수집	<ul style="list-style-type: none">- 공공데이터 포털을 이해한다.- 공개된 파일 형태의 자료 수집 방법을 실습한다.- API 형태로 공개한 JSON 포맷 데이터 수집 방법을 실습한다.- API 형태로 공개한 XML 포맷 데이터 수집 방법을 실습한다.

단원 개요

웹 크롤링



	0	1
0	더마시나 무항생제 구운계란60구, 1개, 2,100g	10,300
1	오복유통 HACCP인증 구운계란 2판60구, 60구, 2판	13,900
2	잡나무촌 무염훈제계란, 30개입, 1.2kg(한판)	6,200
3	꾼란 맥반석 구운계란 30구 1판, 30개입, 1.2kg	7,900
4	맛군 축축 톡톡 구운 계란, 30알, 1박스	7,900
5	감동란 간이베어 있는 축축한 반숙계란, 50g, 30개입	16,900
6	[계란사랑] 맥반석 구운계란 구운란 60구 (2판), 2700g	11,900
7	진주형 오마이 포켓 메주리알 5p, 25g, 10개입	9,440

공공데이터 수집

연도	월	전국 PIR	서울 PIR	부산 PIR	대구 PIR	인천 PIR	광주 PIR	대전 PIR	울산 PIR	대기오염 나쁨 위치		관측소위치	
										0	중구	서울특별시 중구 덕수궁길 15시청서소문별관 3동	
0	2004	3	4.21	4.89	3.95	3.73	4.65	2.81	4.68	2.66	1	청계천로	서울 중구 청계천로 184(청계천4가사거리 남강빌딩 앞)
1	2004	4	4.39	5.59	3.91	3.88	4.59	2.92	3.83	2.74	2	용산구	서울 용산구 한남대로 136서울특별시중부기술교육원
2	2004	5	4.19	5.14	4.90	3.83	4.78	3.41	4.19	2.93	3	강변북로	서울 성동구 강변북로 257한강사업본부 옆
3	2004	6	4.09	4.38	4.20	3.77	4.30	2.83	4.19	2.81	4	홍릉로	서울 동대문구 홍릉로 1(청량리전철역 사거리 SC제일은행 앞)

교육목표: 웹 데이터를 수집하는 방법을 익힌다.

CONTENTS

- 1 웹크롤링 이해 및 기본기 살펴보기
- 2 BeautifulSoup 활용 HTML 추출하기
- 3 태그정보 수집하기
- 4 테이블정보 수집하기
- 5 셀레니움 활용 웹브라우저 자동화



외부연동 데이터 수집 파트1 (웹 크롤링)

모듈 개요

웹 크롤링



Showing (30): All holidays/observances For: 2019
 Jump to: Next [JAN] [FEB] [MAR] [APR] [MAY] [JUN] [JUL] [AUG] [SEP] [OCT] [NOV] [DEC]

Holidays and Observances in South Korea in 2019

Date	Name	Type
1 Jan	Tuesday	New Year's Day
4 Feb	Mo	
5 Feb	Tue	
6 Feb	We	
14 Feb	Th	
1 Mar	Fri	
21 Mar	Th	
5 Apr	Fri	

	DATE	WEEK	HOLIDAY_NAME	HOLIDAY_TYPE
0	1월 1일	월요일	New Year's Day	Public Holiday
1	2월 9일	금요일	2018 Winter Olympics start	Observance
2	2월 14일	수요일	Valentine's Day	Observance
3	2월 15일	목요일	Seollal Holiday	Public Holiday
4	2월 16일	금요일	Seollal	Public Holiday



	0	1
0	더마시나 무항생제 구운계란60구, 1개, 2, 100g	10,300
1	오복유용 HACCP인증 구운계란 2판60구, 60구, 2판	13,900
2	참나무촌 무염혼제계란, 30개입, 1.2kg(한판)	6,200
3	곤란 맥반석 구운계란 30구 1판, 30개입, 1.2kg	7,900
4	맛군 축족 풀깃 구운 계란, 30알, 1박스	7,900
5	감동란 간이버어 있는 축족한 반숙계란, 50g, 30개입	16,900
6	[계란사항] 맥반석 구운계란 구운란 60구 (2판), 2700g	11,900
7	진주형 오마이 포켓 메주리알 5p, 25g, 10개입	9,440

파트1. 웹 크롤링 이해 및 기본기 살펴보기

1. 웹 크롤링 이해 및 기본기 살펴보기

모듈 개요

[과정개요]

웹 크롤링 이해 및 기본기 살펴보기

[교육목표]

- 웹 크롤링 의 필요성을 이해합니다.
- 웹크롤링을 위한 기본적인 문법 및 데이터프레임 생성방법을 실습을 통해 리마인드 합니다.

[교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
웹 크롤링 이해 및 기본기 살펴보기	<ul style="list-style-type: none">- 웹 크롤링이 인공지능 영역에 왜 필요한지 이해합니다.- 기본적인 문법 및 데이터프레임 생성 방법을 실습합니다.

1. 웹 크롤링 이해 및 기본기 살펴보기

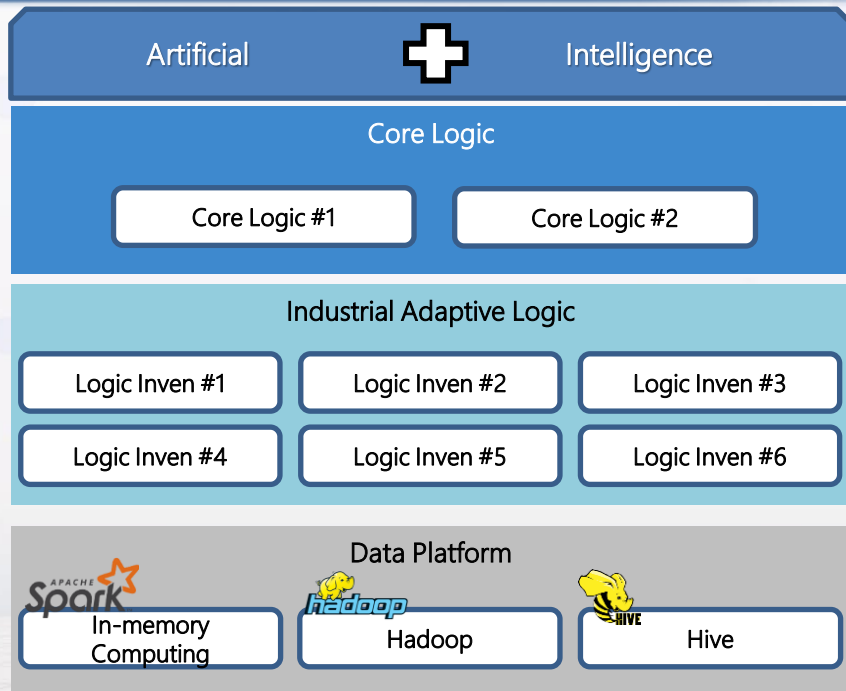
우리 회사에 필요한 가치를 창출하기 위해 구성도를 그린다...

분석 데이터

내부 데이터



이미 사내에서
접속 가능한
정제 후 활용 가능 데이터



시각화 / 웹 시연



Cloud Environment

4차산업혁명 단계별로

www.youtube.com/hkcode

1. 웹 크롤링 이해 및 기본기 살펴보기

시간이 지나자... 조금 한계에 ...



가지고 있는 걸로 분석 해보자..

1일...

2일...

3일...

하아... 가지고 있는 데이터만으로는 다양한 각도로 분석을 할 수 없네.....

1. 웹 크롤링 이해 및 기본기 살펴보기

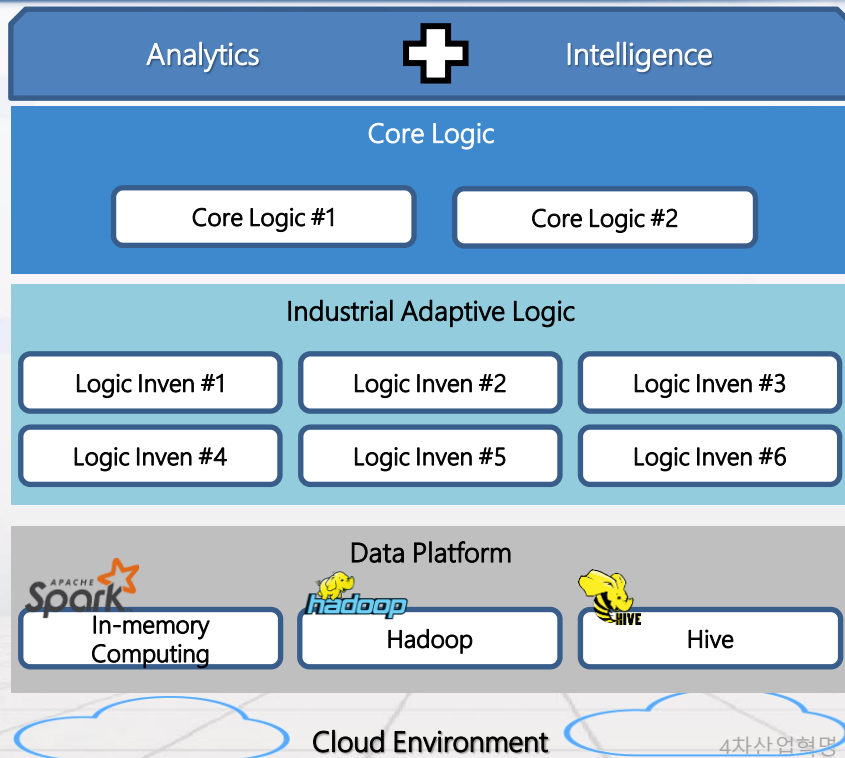
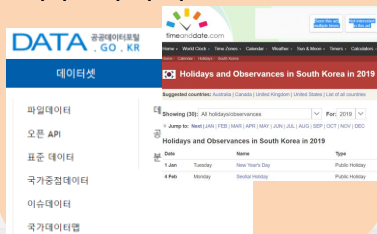
외부에 접속하여 필요한 정보들을 수집해보자!

분석 데이터

내부 데이터



외부 데이터



시각화 / 웹 시연



4차산업혁명 단계별

www.youtube.com/hkcode

1. 웹 크롤링 이해 및 기본기 살펴보기

데이터가 많아지니 분석해볼만한게 많아지네!

웹 상에 공개된 데이터 스크랩 후 활용

접속 사이트	제공항목(데이터셋)
국가통계포털	코스닥지수
국가통계포털	코스닥 150 지수
국가통계포털	코스닥 주가이익비율 (PER)
국가통계포털	코스닥 주가순자산비율 (PBR)
국가통계포털	코스닥 배당수익률
국가통계포털	코스닥 산업별 투자지표
공공데이터포털	한국감정원 오피스텔 동향조사 현황
공공데이터포털	한국감정원 부동산 매매가격지수 현황
공공데이터포털	공간융합정보

내부 데이터 정제 후 활용



데이터 탐색 후 필요 데이터 직접 스크랩 후 활용



앞으로 웹에서 데이터를 가져오는 방법을 배웁니다.
그전에! **핵심문법**을 둘러보겠습니다.
(**이미 배웠다면 Pass!!!**)



Python 기본기 살펴보기

김효관 |

1. 웹 크롤링 이해 및 기본기 살펴보기

주요 문법

프로그래밍의 문법을 알면 다양한 각도로 문제를 해결할 수 있다.

1 반복하기

for / while

2 조건 판단하기

if

3 자주사용하는내용 함수화 하기

def testFunction(inVal) :

1. 웹 크롤링 이해 및 기본기 살펴보기

1. 반복하기 (for, while)



```
tvList = [ UN40EN001, UN40EN002, UN40EN003, UN40EN004]
```

헉! tv목록앞에 제품목록을 전부 붙여야하는데.... 어떻게하지?
하나씩 하면 되겠네..

```
preFix = "LEDTV_"  
tvList[0] = prefix + tvList[0]  
tvList[1] = prefix + tvList[1]
```

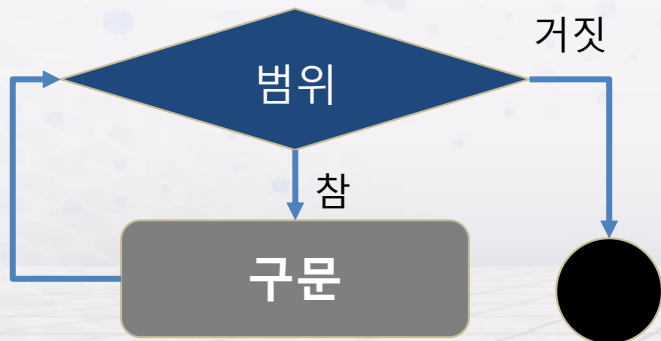
만약.. 목록이 십만개가 넘는다면??

1. 웹 크롤링 이해 및 기본기 살펴보기

1. 반복하기 (for, while)

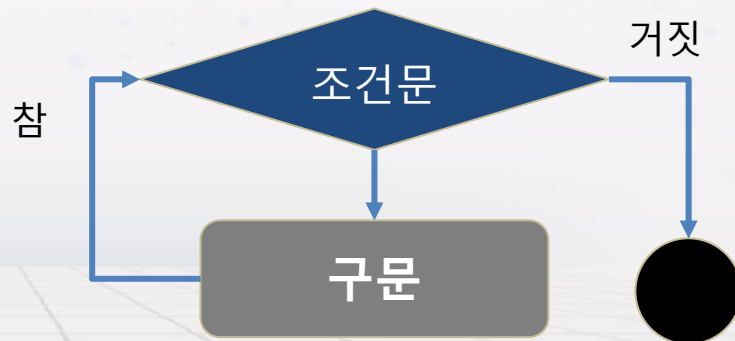
for

주어진 범위가 끝날때까지 구문 수행



while

조건문이 참인 경우 구문 수행
거짓인 경우 끝



1. 웹 크롤링 이해 및 기본기 살펴보기

참고. 비교연산자의 이해 및 논리

비교 연산자	설 명	예 시
$A == B$	A와 B가 같으면 참, 그렇지 않으면 거짓	$A == B$
$A != B$	A와 B가 다르면 참, 그렇지 않으면 거짓	$A != B$
$A > B$	A가 B보다 크면 참, 그렇지 않으면 거짓	$A > B$
$A < B$	A가 B보다 작으면 참, 그렇지 않으면 거짓	$A < B$
$A >= B$	A가 B보다 크거나 같으면 참, 그렇지 않으면 거짓	$A >= B$
$A <= B$	A가 B보다 작거나 같으면 참, 그렇지 않으면 거짓	$A <= B$

논리 연산자	설 명	예 시
$A \& B$	A 와 B가 모두참인경우 참	A 및 B 기능을 만족
$A B$	A 와 B중 하나만 참이면 참	A 또는 B 기능을 만족
$\sim A$	A가 아님	A를 제외한 항목

1. 웹 크롤링 이해 및 기본기 살펴보기

1. 반복하기 (for, while)

while (조건) loop : 조건 실패 시 조건 문 탈출
for (조건) : 조건 실패 시 조건 문 탈출

Example

줄 마지막에 백슬러시"\" 사용 시 줄이음 가능

```
tvList = [ 'UN40EN001', 'UN40EN002', 'UN40EN003', 'UN40EN004']
```

for 문

```
preFix = "LEDTV_"  
### list(range(0, 4, 1))  
for i in range(0, 4, 1):  
    tvList[i] = preFix + tvList[i]
```

범위/로직 구문을 위한
탭공백!

for문은 변수가 특정 범위안에서
구문연산 (구문은 탭으로 한칸땀)

while 문

```
preFix = "LEDTV_"  
  
i=0  
listLength = len(tvList)  
while(i < listLength ):  
    tvList[i] = preFix + tvList[i]  
    i = i+1
```

while문은 조건식을
만족하는 동안 구문연산

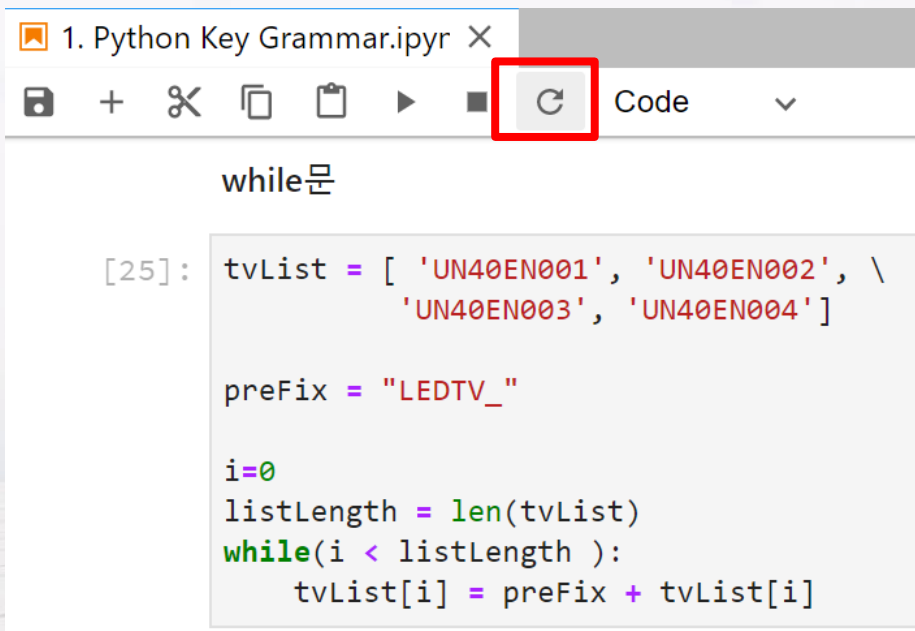
빠지면 무한 루프!

4차

1. 웹 크롤링 이해 및 기본기 살펴보기

1. 반복하기 (while, for)

while문에서 무한 루프에 빠지면?



1. Python Key Grammar.ipynr X

Run + Copy Paste Run and Cell Run and All Code

```
while문

[25]: tvList = [ 'UN40EN001', 'UN40EN002', \
                'UN40EN003', 'UN40EN004' ]

preFix = "LEDTV_"

i=0
listLength = len(tvList)
while(i < listLength ):
    tvList[i] = preFix + tvList[i]
```

```
tvList2 = [ 'UN40EN001', 'UN40EN002',  
            'UN40EN003', 'UN40EN004']  
리스트가 출력되도록 하세요
```

힌트: `print(tvList[0])`

1. 웹 크롤링 이해 및 기본기 살펴보기

2. 조건 판단하기



```
tvList = [ UN40EN001, LEDTV_UN40EN002,  
           LEDTV_UN40EN003, UN40EN004]
```

헉! tv목록앞에 제품목록을 전부 붙여야하는데.... 붙어있는게 있고 없는게 있네.. 어떻게 하지?
붙어 있는지 조건을 판단해야겠는데...

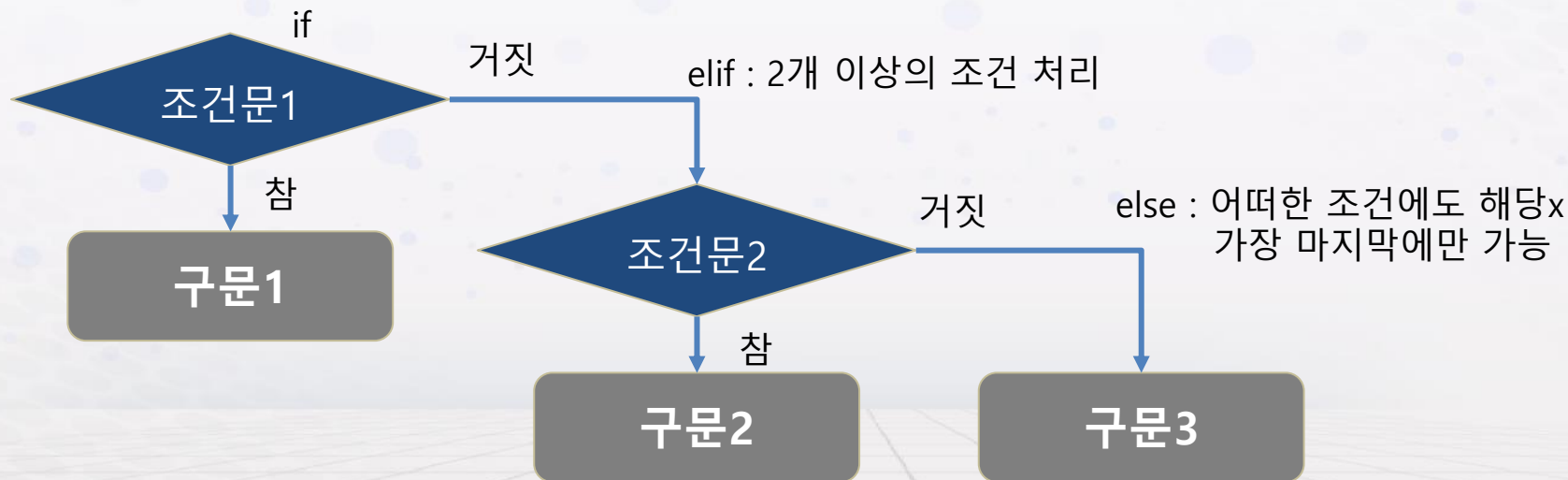
(만약, LEDTV_가 있으면 넘기고... LEDTV_가 없으면 넣고..)

조건문은 어떻게 사용할 수 있을까요?

1. 웹 크롤링 이해 및 기본기 살펴보기

2. 조건 판단하기

조건문을 평가하고 참인 경우만 구문 수행



1. 웹 크롤링 이해 및 기본기 살펴보기

참고. 비교연산자의 이해 및 논리

비교 연산자	설 명	예 시
$A == B$	A와 B가 같으면 참, 그렇지 않으면 거짓	$A == B$
$A != B$	A와 B가 다르면 참, 그렇지 않으면 거짓	$A != B$
$A > B$	A가 B보다 크면 참, 그렇지 않으면 거짓	$A > B$
$A < B$	A가 B보다 작으면 참, 그렇지 않으면 거짓	$A < B$
$A >= B$	A가 B보다 크거나 같으면 참, 그렇지 않으면 거짓	$A >= B$
$A <= B$	A가 B보다 작거나 같으면 참, 그렇지 않으면 거짓	$A <= B$

논리 연산자	설 명	예 시
$A \& B$	A 와 B가 모두참인경우 참	A 및 B 기능을 만족
$A B$	A 와 B중 하나만 참이면 참	A 또는 B 기능을 만족
$\sim A$	A가 아님	A를 제외한 항목

1. 웹 크롤링 이해 및 기본기 살펴보기

2. 조건 판단하기

```
if [조건문]:  
    [조건문 참인경우 실행]  
else:  
    [조건문 불일치 시 실행]
```

Example

```
testModel = "UN40EN001"  
preFix = "LEDTV_"  
  
if testModel.count(preFix) == 0:  
    testModel = preFix + testModel  
else:  
    pass
```

```
testModel = "UN40EN001"  
preFix = "LEDTV_"  
  
if testModel.count(preFix) == 0:  
    testModel = preFix + testModel  
else:  
    pass
```

testModel

'LEDTV_UN40EN001'


```
nationList ['A01' , '한국' , 'A02' , '미국' , 'A03' , '프랑스']  
국가코드만 출력하세요
```

1. 웹 크롤링 이해 및 기본기 살펴보기

Pandas DataFrame

1 고성능 데이터 조작 라이브러리

2 스프레드시트, RDB 데이터 접근

3 고성능 시계열 처리 기능

`pip install pandas`

<https://pandas.pydata.org/>

```
import pandas as pd
```

```
# 리스트 활용 데이터프레임 생성
```

```
testDf = pd.DataFrame(  
    [ ["A01", "201501", 100],  
      ["A02", "201502", 200] ] )
```

```
# testDf.head()
```

```
# 컬럼명 변경
```

```
testDf.rename(columns={0:"regionid",  
                       1:"yearweek",  
                       2:"sales"})
```

	regionid	yearweek	sales
0	A01	201501	100
1	A02	201502	200

1. 웹 크롤링 이해 및 기본기 살펴보기

Pandas DataFrame

행/열 구조의 가장 많이 활용되는 자료구조 형태

Example

데이터 프레임 라이브러리 활용

```
import pandas as pd
```

딕셔너리를 활용한 데이터프레임 생성

```
data = {'name': ['A고객', 'B고객', 'C고객', 'D고객'],  
        'age': [27, 40, 33, 29],  
        'stock_age': [2, 10, 5, 1]}
```

```
dataFrame = pd.DataFrame(data)
```

데이터 프레임 : 스프레드시트 형태의 자료구조

```
print(dataFrame)
```

6. Pandas Dataframe

데이터 프레임 라이브러리 활용

```
import pandas as pd
```

딕셔너리를 활용한 데이터프레임 생성

```
data = {'name': ['A고객', 'B고객', 'C고객', 'D고객'],  
        'age': [27, 40, 33, 29],  
        'stock_age': [2, 10, 5, 1]}
```

```
dataFrame = pd.DataFrame(data)
```

데이터 프레임 : 스프레드시트 형태의 자료구조

```
print(dataFrame)
```

	name	age	stock_age
0	A고객	27	2
1	B고객	40	10
2	C고객	33	5
3	D고객	29	1

1. 웹 크롤링 이해 및 기본기 살펴보기

Pandas DataFrame

리스트를 활용한 데이터 분석을 위한 자료구조 생성

Example

```
import pandas as pd
# 리스트 생성
test = [10,100,1000,10000]
# 데이터프레임 변환
testDf = pd.DataFrame(test)
testDf

testDf.columns=["test"]
testDf
```

2. 리스트를 활용한 Pandas DataFrame 생성

```
import pandas as pd
# 리스트 생성
test = [10,100,1000,10000]
# 데이터프레임 변환
testDf = pd.DataFrame(test)
testDf
```

	0
0	10
1	100
2	1000
3	10000

```
testDf.columns=["test"]
testDf
```

	test
0	10
1	100
2	1000
3	10000

1. 웹 크롤링 이해 및 기본기 살펴보기

Pandas DataFrame

리스트를 활용한 데이터 분석을 위한 자료구조 생성

Example

```
import pandas as pd
date = ['16.02.29', '16.02.26', '16.02.25', '16.02.24', '16.02.23']
date2 = ['17.02.29', '17.02.26', '17.02.25', '17.02.24', '17.02.23']
```

```
date_df = pd.DataFrame(date, columns=['date2'])
date_df2 = pd.DataFrame(date2, columns=['date23'])
```

```
final = pd.concat([date_df, date_df2], axis = 1)
final
```

* zip 활용

```
import pandas as pd
date = ['16.02.29', '16.02.26', '16.02.25', '16.02.24', '16.02.23']
date2 = ['17.02.29', '17.02.26', '17.02.25', '17.02.24', '17.02.23']
```

```
date_df = pd.DataFrame(date, columns=['date2'])
date_df2 = pd.DataFrame(date2, columns=['date23'])
```

```
final = pd.concat([date_df, date_df2], axis = 1)
final
```

	date2	date23
0	16.02.29	17.02.29
1	16.02.26	17.02.26
2	16.02.25	17.02.25
3	16.02.24	17.02.24
4	16.02.23	17.02.23

파트2. BeautifulSoup 활용 HTML 추출하기

2. BeautifulSoup 활용 HTML 추출하기

모듈 개요

[과정개요]

웹 소스 접근방법 및 소스코드 추출하기

[교육목표]

- 웹 페이지 소스 접근방법을 실습합니다.
- BeautifulSoup를 활용하여 태그 소스 추출 방법 실습합니다.

[교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
BeautifulSoup 활용 HTML 추출하기	<ul style="list-style-type: none">- Requests 라이브러리 활용 페이지 소스 가져오기- BeautifulSoup 활용 HTML 태그 구조 추출하기

2. BeautifulSoup 활용 HTML 추출하기

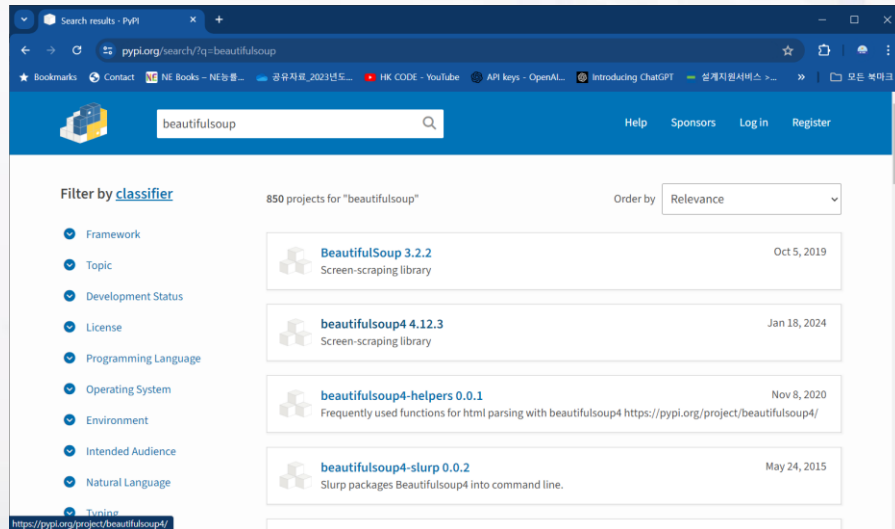
데이터 불러오기 from web (BeautifulSoup 라이브러리)

1 웹 사이트 데이터 추출 라이브러리

2 HTML/XML 정보 추출

pip install bs4

HTML: Hyper Text Markup Language
- 웹페이지를 표시하는 HTML은 마크업언어이다
태그, 요소, 속성 등의 구성요소를 이용해
문서를 표현한다. <html> </html>



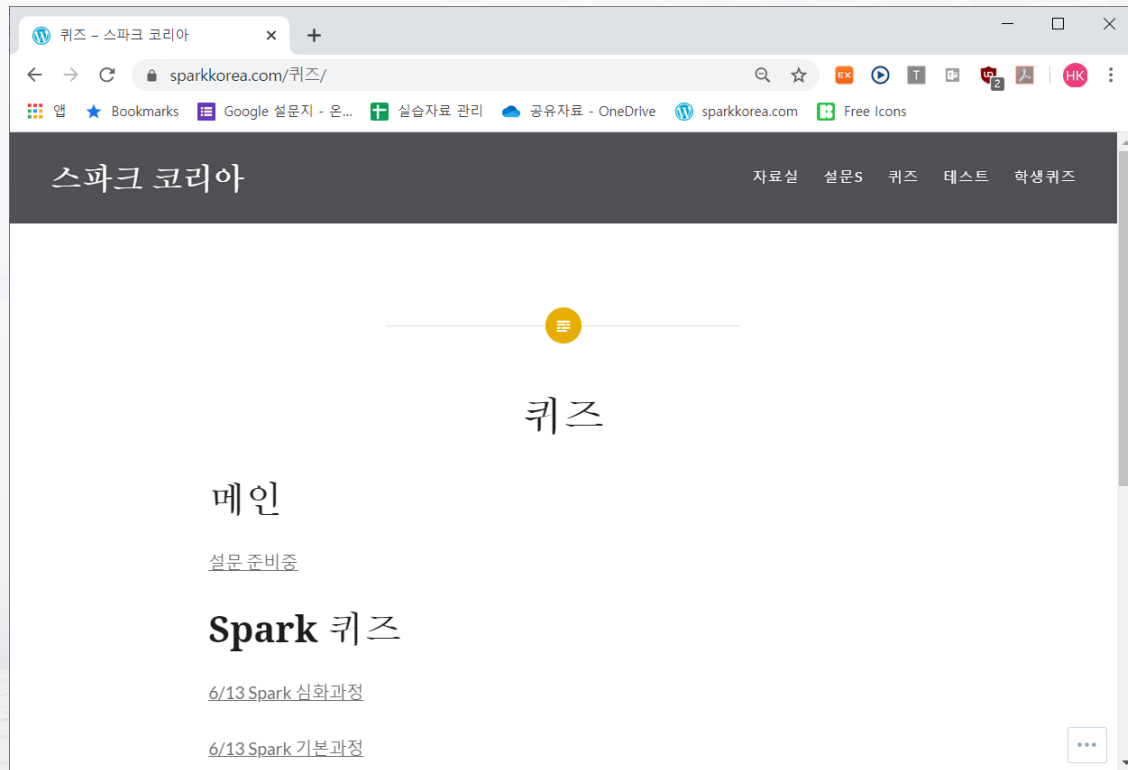
<https://pypi.org/> (파이썬 패키지 인덱스)

2. BeautifulSoup 활용 HTML 추출하기

Mission

<https://sparkkorea.com/quiz>

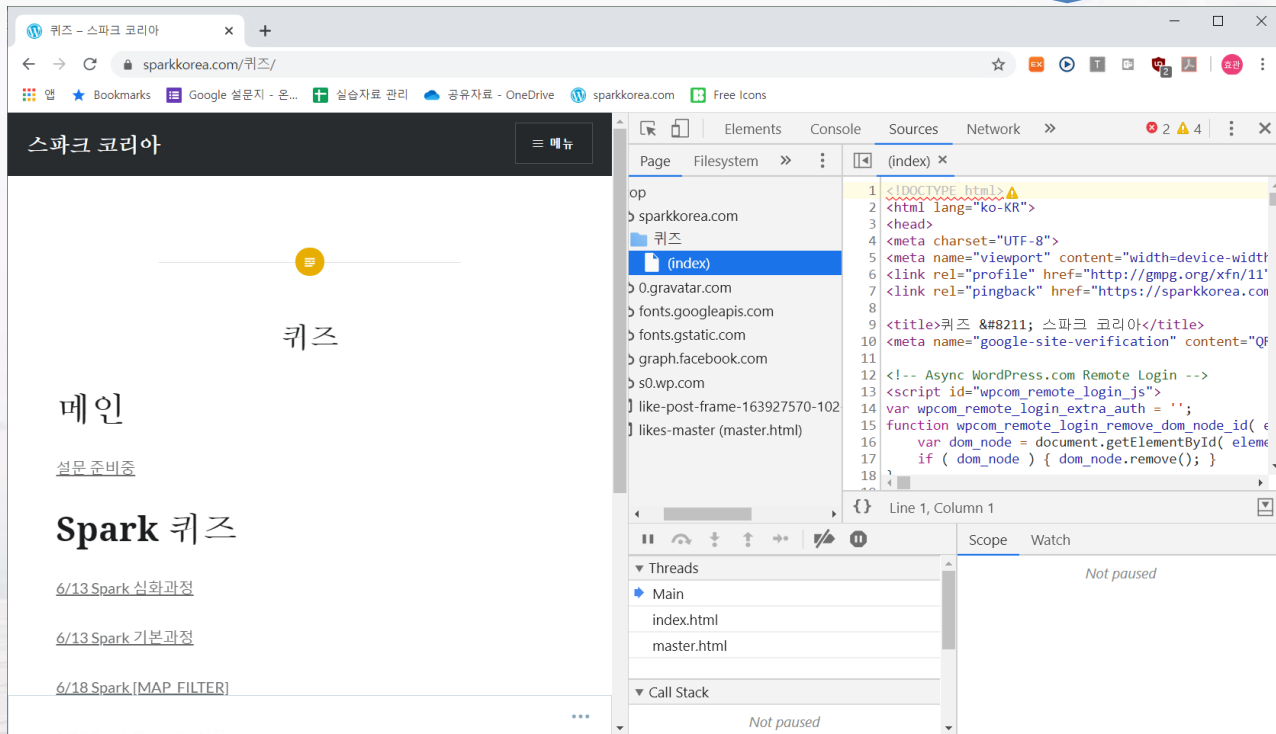
사이트에 접속 후 퀴즈 내용을
스크랩 하자!



2. BeautifulSoup 활용 HTML 추출하기

1. 웹페이지 이해

크롬 브라우저 실행
F12 버튼 눌러서 개발자도구 진입 후 Source (index) 클릭



2. BeautifulSoup 활용 HTML 추출하기

1. 웹페이지 이해



2. BeautifulSoup 활용 HTML 추출하기

2. 라이브러리 선언 및 html 소스 불러오기

```
import requests, bs4
```

크롤링 대상 URL

```
# 웹페이지 요청 (200: 정상) 및 소스 가져오기  
resp = requests.get("https://sparkkorea.com/퀴즈/")  
html = resp.text
```

```
# 태그정보 이쁘게 추출  
bs = bs4.BeautifulSoup(html, 'html.parser')
```

HTML 태그 추출

2. BeautifulSoup 활용 HTML 추출하기

2. 라이브러리 선언 및 html 소스 불러오기

원 소스전부 스크랩

```
html = resp.text
```

html parser 활용한 데이터 정제

```
bs = bs4.BeautifulSoup(html, 'html.parser')
```

태그값을 가져온다

Example

```
import requests, bs4  
import pandas as pd
```

라이브러리 선언

#웹페이지 html 소스 가져오기

```
resp = requests.get("https://sparkkorea.com/퀴즈/")
```

대상 url 소스요청

```
resp.encoding='utf-8'
```

```
html = resp.text
```

```
bs = bs4.BeautifulSoup(html, 'html.parser')
```

bs4를 활용한 html DOM구조만 남김

```
bs
```

2. BeautifulSoup 활용 HTML 추출하기

2. 라이브러리 선언 및 html 소스 불러오기

```
import requests, bs4
import pandas as pd

#웹페이지 html 소스 가져오기
resp = requests.get("https://sparkkorea.com/퀴즈/")
resp.encoding='utf-8'
html = resp.text
```

```
bs = bs4.BeautifulSoup(html, 'html.parser')
bs
```

```
<!DOCTYPE html>
```

```
<html lang="ko-KR">
```

```
<head>
```

```
<meta charset="utf-8"/>
```

```
<meta content="width=device-width, initial-scale=1" name="viewport"/>
```

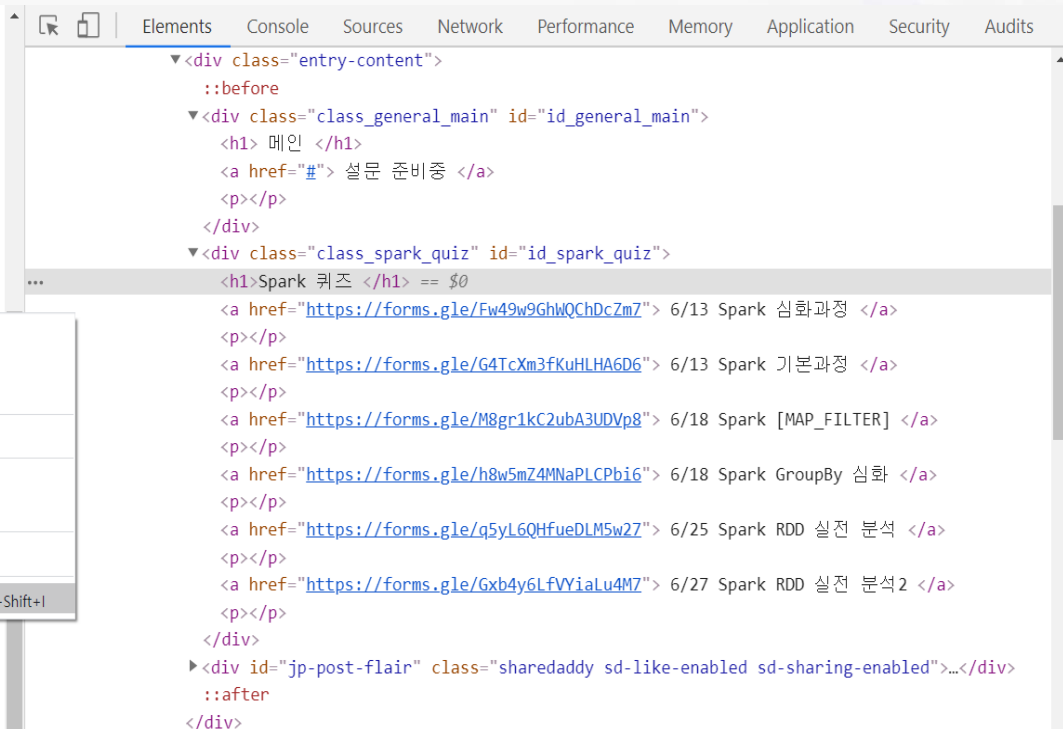
```
<link href="http://gmpg.org/xfn/11" rel="profile"/>
```

```
<link href="https://sparkkorea.com/xmlrpc.php" rel="pingback"/>
```

```
<title>퀴즈 - 스파크 코리아</title>
```

2. BeautifulSoup 활용 HTML 추출하기

팁. 원하는 정보 소스로 바로 이동



파트3. 태그정보 수집하기

3. 태그정보 수집하기

모듈 개요

[과정개요]

태그정보 수집하기

[교육목표]

- 페이지 소스 내 원하는 태그정보를 수집하는 방법을 이해하고 실습합니다.

[교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
태그정보 수집하기	- 원하는 웹페이지 소스를 수집한 후 필요한 태그 정보 내 소스만 탐색하는 방법을 실습을 통해 학습합니다.

3. 태그정보 수집하기

1. 태그명으로 소스 수집하기

```
bs = bs4.BeautifulSoup(html, "html.parser")
# div 태그 찾기
divs = bs.find or findAll("태그명")
```

태그 찾기 (맨 앞 한개)

bs.find(name = "태그명")

태그 찾기 (전체)

bs.findAll(name = "태그명")

a 태그

```
<!doctype html>
<html lang="ko">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>스파크 코리아</title>
  </head>
  <body>
    <div id="page" class="wp-embed-responsive customizer-highlander-enabled mgnia-right not-scrolled">
      <a class="skip-link screen-reader-text" href="#content">컨텐츠로 건너뛰기</a>
      <header id="masthead" class="site-header" role="banner">
        <div class="site-branding">
          <h1 class="site-title">스파크 코리아</h1>
          <p class="site-description">스파크코리아 자료공유</p>
        </div>
        <!-- .site-branding -->
      </header>
      <nav id="site-navigation" class="main-navigation" role="navigation">...</nav>
    </body>
  </html>
```

```
# html 태그 정보 가져오기
aTag = bs.find(name = "a")
aTag
```

컨텐츠로 건너뛰기

```
# html 태그 정보 가져오기
aTags = bs.findAll(name = "a", limit=2)
aTags
```

```
[<a class="skip-link screen-reader-text" href="#content">컨텐츠로 건너뛰기</a>,
 <a href="https://sparkkorea.com/" rel="home">
```

스파크 코리아

] 교수

3. 태그정보 수집하기

1. 태그명으로 소스 수집하기

```
bs = bs4.BeautifulSoup(html, "html.parser")
# div 태그 찾기
divs = bs.find or findAll(name = "태그명")
```

find: 첫번째 태그정보 리턴
findAll: 조건에 해당되는 모든 태그를
리스트로 리턴

Example

코드 내 a 태그 1개만 탐색

```
bs = bs4.BeautifulSoup(html, "html.parser")
aTag = bs.find(name = "a")
print(aTag)
print( type(aTag) )
```

코드 내 a 태그 상단 2개만 탐색

```
aTags = bs.findAll(name = "a", limit = 3)
print(aTags)
print( type(aTags) )
```

```
print(aTag.text)
print(aTag.name)
```

```
# 코드 내 a 태그 1개만 탐색
bs = bs4.BeautifulSoup(html, "html.parser")
aTag = bs.find(name = "a")
print(aTag)
print( type(aTag) )
# 코드 내 a 태그 탐색, 상단 2개만
aTags = bs.findAll(name = "a", limit = 3)
print(aTags)
print( type(aTags) )
```

```
<a class="skip-link screen-reader-text" href="#content">컨텐츠로 건너뛰기</a>
<class 'bs4.element.Tag'>
[<a class="skip-link screen-reader-text" href="#content">컨텐츠로 건너뛰기</a>, <a href="http
s://sparkkorea.com/" rel="home">
스파크 코리아
=>"https://sparkkorea.com/%ec%9e%90%eb%a3%8c%ec%8b%a4/">자료실</a>]
<class 'bs4.element.ResultSet'>
```

```
print(aTag.text)
print(aTag.name)
```

컨텐츠로 건너뛰기

a

3. 태그정보 수집하기

2. 태그 속성정보로 소스 수집하기

```
bs = bs4.BeautifulSoup(html, "html.parser")  
# "id": "id 명" or "class": "class명"  
findAttr = bs.findAll or find(name = "태그명", attrs = {"속성명[id, class, etc,..]": "속성이름"})
```

```
<div class="class_spark_quiz" id="id_spark_quiz">
```

```
<div class="class_spark_quiz" id="id_spark_quiz">  
<h1>Spark 퀴즈 </h1>  
<a href="https://forms.gle/Fw49w9GhwQChDcZmZ"> 6/13 Spark 심화과정 </a>  
<p></p>  
<a href="https://forms.gle/G4TcXm3fKuHLHA6D6"> 6/13 Spark 기본과정 </a>  
<p></p>  
<a href="https://forms.gle/M8gr1kC2ubA3UDVp8"> 6/18 Spark [MAP_FILTER] </a>  
<p></p>  
<a href="https://forms.gle/h8w5mZ4MNaPlCPbi6"> 6/18 Spark GroupBy 심화 </a>  
<p></p>  
<a href="https://forms.gle/q5yL6QHfueDLM5w27"> 6/25 Spark RDD 실전 분석 </a>  
<p></p>  
<a href="https://forms.gle/Gxb4y6LfVYiaU4M7"> 6/27 Spark RDD 실전 분석2 </a>  
<p></p>  
</div>
```



단락을 나타내는 div 태그가 여러개인데...
원하는 단락만 어떻게 접근하지?

태그 찾기 (맨 앞 한개)

태그 찾기 (전체)

```
bs.find( name = "태그명",  
        attrs = { "속성명": "속성이름" } )
```

4차산업혁명 단계별 교육은 빅데이터&인공지능(공민각, 김호관 교수)

www.youtube.com/hkcode

3. 태그정보 수집하기

2. 태그 속성정보로 소스 수집하기

```
bs = bs4.BeautifulSoup(html, "html.parser")
# "id": "id 명" or "class": "class명 "
findAttr = bs.findAll or find(name = "태그명", attrs = {"속성명[id, class, etc,...]": "속성이름"})
```

Example

```
# ID 속성으로 태그 탐색
spQuizTag = bs.find(name = "div",
                    attrs = {"id": "id_spark_quiz"})
spQuizTag
```

```
bs = bs4.BeautifulSoup(html, "html.parser")
# ID 속성으로 태그 탐색
spQuizTag = bs.find(name = "div",
                    attrs = {"id": "id_spark_quiz"})
spQuizTag

<div class="class_spark_quiz" id="id_spark_quiz">
<h1>Spark 퀴즈 </h1>
<a href="https://forms.gle/Fw49w9GhWQChDcZm7"> 6/13 Spark 심화과정 </a>
<p></p>
<a href="https://forms.gle/G4TcXm3fKuHLHA6D6"> 6/13 Spark 기본과정 </a>
<p></p>
<a href="https://forms.gle/M8gr1kC2ubA3UDVp8"> 6/18 Spark [MAP_FILTER] </a>
<p></p>
<a href="https://forms.gle/h8w5mZ4MNaPLCPbi6"> 6/18 Spark GroupBy 심화 </a>
<p></p>
<a href="https://forms.gle/q5yL6QHfueDLM5w27"> 6/25 Spark RDD 실전 분석 </a>
<p></p>
<a href="https://forms.gle/Gxb4y6LfVYiaLu4M7"> 6/27 Spark RDD 실전 분석2 </a>
<p></p>
</div>
```

3. 태그정보 수집하기

3. 태그 내 부분태그 소스 수집하기

```
bs = bs4.BeautifulSoup(html, "html.parser")
findAttr = bs.findAll or find(name = "태그명", attrs = {"속성명": "속성이름"})
# find내에 tag명으로 다시 찾기
findPart = findAttr.find(name = "a")
```



태그 소스가 너무 광범위해서 필요한
태그 내 소스에서 태그를 더 찾을 수 없을까?

처음 찾은 태그

`findAttr = bs.find(name = "태그명")`

처음 찾은 태그 소스에서 태그 검색

`findPart = findAttr.find(name = "태그명")`

다시 작업하면 단계별로 원하는 빅데이터&인공지능 관련 강의를 검색할 수 있습니다.
www.youibook.com/chkcode

```
<div class="class_spark_quiz" id="id_spark_quiz">
  <h1>Spark 퀴즈 </h1>
  <a href="https://forms.gle/Fw49w9GhWQChDc7mZ"> 6/13 Spark 심화과정 </a>
<p></p>
  <a href="https://forms.gle/G4TcXm3fKuHLHA6D6"> 6/13 Spark 기본과정 </a>
<p></p>
  <a href="https://forms.gle/M8gr1kC2ubA3UDVp8"> 6/18 Spark [MAP_FILTER] </a>
<p></p>
  <a href="https://forms.gle/h8w5mZ4MNaPLCPbi6"> 6/18 Spark GroupBy 심화 </a>
<p></p>
  <a href="https://forms.gle/q5yL6QHfueDLM5w27"> 6/25 Spark RDD 실전 분석 </a>
<p></p>
  <a href="https://forms.gle/Gxb4y6LfVYiaLu4M7"> 6/27 Spark RDD 실전 분석2 </a>
<p></p>
</div>
```

3. 태그정보 수집하기

3. 태그 내 부분태그 소스 수집하기

```
bs = bs4.BeautifulSoup(html, "html.parser")
findAttr = bs.findAll or find(name = "태그명", attrs = {"속성명": "속성이름"})
# find내에 tag명으로 다시 찾기
findPart = findAttr.find(name = "a")
```

Example

```
# html 부분구조 가져오기
spQuizTagLink = spQuizTag.find(name = "a")
spQuizTagLink
```

```
# html 부분구조 가져오기
spQuizTagLink = spQuizTag.find(name = "a")
spQuizTagLink
```

```
<a href="https://forms.gle/Fw49w9GhWQChDcZm7"> 6/13 Spark 심화과정 </a>
```

3. 태그정보 수집하기

4. 태그 내 속성정보 수집하기

```
bs = bs4.BeautifulSoup(html, "html.parser")
findAttr = bs.findAll or find("태그명", {"속성명": "속성이름"})
# 속성정보 찾기
print(findPart.find('a').attrs['href'])
```

태그 내 text 속성

 6/13 Spark 심화과정

href 속성



태그 내 속성정보를 수집하는 방법은?

findAttr = findPart.attrs["속성명"] ← 속성 찾기

findText = findPart.text ← 태그 텍스트 찾기

findTag = findPart.name ← 태그명 찾기

3. 태그정보 수집하기

4. 태그 내 속성 정보 가져오기

```
bs = bs4.BeautifulSoup(html, "html.parser")
findAttr = bs.findAll or find("태그명", {"속성명": "속성이름"})
# 속성정보 찾기
print(findPart.find('a').attrs['href'])
```

Example

```
# html 부분구조 가져오기
spQuizTagLink = spQuizTag.find(name = "a")
spQuizTagLink

# 링크 속성정보 가져오기
linkAttrs = spQuizTagLink.attrs['href'] --- (1)
linkText = spQuizTagLink.text
linkTag = spQuizTagLink.name
print(linkAttrs, linkText, linkTag)
```

```
# html 부분구조 가져오기
spQuizTagLink = spQuizTag.find(name = "a")
spQuizTagLink
# 링크 속성정보 가져오기
linkAttrs = spQuizTagLink.attrs['href']
linkText = spQuizTagLink.text
linkTag = spQuizTagLink.name
print(linkAttrs, linkText, linkTag)
```

<https://forms.gle/Fw49w9GhWQChDcZm7> 6/13 Spark 심화과정 a

3. 태그정보 수집하기

참조. findAll 디버깅하기

```
bs = bs4.BeautifulSoup(html, "html.parser")
findAttr = bs.findAll or find("태그명", {"속성명": "속성이름"})
# 속성정보 찾기
print(findPart.find('a').attrs['href'])
```

Example

```
spQuizTag = bs.find(name = "div",
    attrs = {"id": "id_spark_quiz"} )
```

html 부분구조 가져오기

```
spQuizTagLinks = spQuizTag.findAll(name = "a")
spQuizTagLinks
```

findAll -> list 반환

```
spQuizTagLinks[0]
print(spQuizTagLinks[0].text)
print(spQuizTagLinks[0].attrs["href"])
# 그리고 반복문 수행
# for divLink in a_links:
```

3. 태그정보 수집하기

참조. List 활용 데이터프레임 만들기

행 (리스트 데이터) 활용 데이터 프레임 생성

Example

```
import pandas as pd

testList = [ ["링크1_링크","링크1_타이틀"],
             ["링크2_링크","링크2_타이틀"] ]

pd.DataFrame(testList, columns = ["링크","타이틀"])
```

```
import pandas as pd

testList = [ ["링크1_링크","링크1_타이틀"],
             ["링크2_링크","링크2_타이틀"] ]

pd.DataFrame(testList, columns = ["링크","타이틀"])
```

	링크	타이틀
0	링크1_링크	링크1_타이틀
1	링크2_링크	링크2_타이틀

3. 태그정보 수집하기

참조. List 활용 데이터프레임 만들기

컬럼 (리스트 데이터) 활용 데이터 프레임 생성

Example

Import pandas as pd

```
linkColumn = ["링크1_링크", "링크2_링크"]  
titleColumn = ["링크1_타이틀", "링크2_타이틀"]  
pd.DataFrame({"링크":linkColumn,"제목":titleColumn})
```

```
pd.DataFrame(zip(linkColumn,titleColumn),  
             columns = ["링크","제목"])
```

```
import pandas as pd
```

```
linkColumn = ["링크1_링크", "링크2_링크"]  
titleColumn = ["링크1_타이틀", "링크2_타이틀"]  
pd.DataFrame({"링크":linkColumn,"제목":titleColumn})  
pd.DataFrame(zip(linkColumn,titleColumn),  
             columns = ["링크","제목"])
```

	링크	제목
0	링크1_링크	링크1_타이틀
1	링크2_링크	링크2_타이틀

[find 활용] sparkkorea.com 사이트내 퀴즈 페이지 에서
스파크 퀴즈 퀴즈이름 및 링크정보를 스크랩 하세요

	spark퀴즈 타이틀	spark퀴즈 링크
0	6/13 Spark 심화과정	https://forms.gle/Fw49w9GhWQChDcZm7
1	6/13 Spark 기본과정	https://forms.gle/G4TcXm3fKuHLHA6D6
2	6/18 Spark [MAP_FILTER]	https://forms.gle/M8gr1kC2ubA3UDVp8
3	6/18 Spark GroupBy 심화	https://forms.gle/h8w5mZ4MNaPLCPbi6
4	6/25 Spark RDD 실전 분석	https://forms.gle/q5yL6QHfueDLM5w27
5	6/27 Spark RDD 실전 분석2	https://forms.gle/Gxb4y6LfVVYiaLu4M7

파트4. 테이블정보 수집하기

4. 테이블정보 수집하기

모듈 개요

[과정개요]

테이블정보 수집하기

[교육목표]

- 페이지 소스 내 원하는 테이블 정보를 수집하는 방법을 이해하고 실습합니다.

[교육대상]

- 데이터 분석가 / 인공지능 전문가
- 데이터 엔지니어

내용	학습내용
테이블정보 수집하기	- 원하는 웹페이지 소스를 수집한 후 정리가 잘되어 있는 테이블 정보를 수집한다.

4. 테이블정보 수집하기

테이블 데이터 크롤링 (www.sparkkorea.com/테스트)

테스트

편집

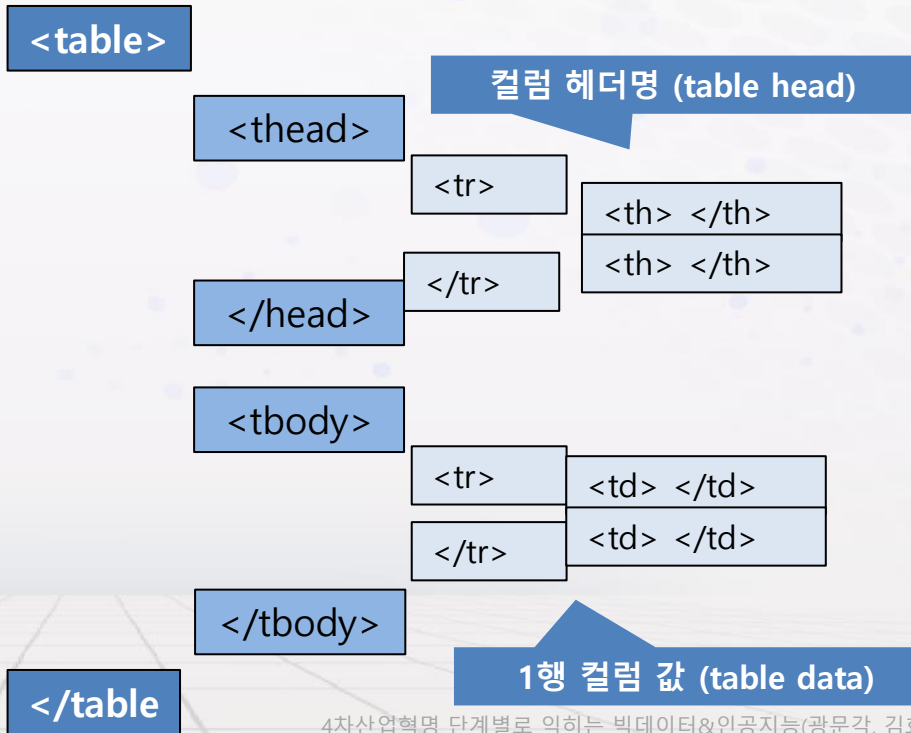
학번	이름
101	김효관
102	이순신
103	김어진

```
<table id="test_table" class="type07"> == $0
  <thead>
    <tr>
      <th scope="cols">학번</th>
      <th scope="cols">이름</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>101</td>
      <td>김 효관</td>
    </tr>
    <tr>
      <td>102</td>
      <td>이순신</td>
    </tr>
    <tr>
      <td>103</td>
      <td>김어진</td>
    </tr>
  </tbody>
</table>
```


4. 테이블정보 수집하기

테이블트리 이해

```
<table id= test_table" class="type07"> == $0
▼<thead>
  ▼<tr>
    <th scope="cols">학번</th>
    <th scope="cols">이름</th>
  </tr>
</thead>
▼<tbody>
  ▼<tr>
    <td>101</td>
    <td>김효관</td>
  </tr>
  ▼<tr>
    <td>102</td>
    <td>이순신</td>
  </tr>
  ▼<tr>
    <td>103</td>
    <td>김어진</td>
  </tr>
</tbody>
</table>
```



4. 테이블정보 수집하기

테이블트리 크롤링 전략

학번	이름
101	김효관
102	이순신
103	김어진



전체를 데이터 담을 리스트!



한 행의 컬럼값 담을 리스트

```
[<tr>
  <td>101</td>
  <td>김효관</td>
</tr>, <tr>
  <td>102</td>
  <td>이순신</td>
</tr>, <tr>
  <td>103</td>
  <td>김어진</td>
</tr>]
```

1행 값 담을 리스트

2행 값 담을 리스트

3행 값 담을 리스트

1. 테이블을 찾는다.
2. 모든 행 데이터를 찾는다 (tbody의 tr태그를 찾는다)
3. 행을 반복하면서 컬럼 데이터를 수집한다.
 - 반복하면서 컬럼의 모든 값을 빈 리스트에 담는다.
columnList = 컬럼1 + 컬럼2 + ..
4. 한 행의 모든 컬럼을 담았으면 전체 행을 리스트에 추가한다.
 - rowList.append(columnList)
5. columnList를 초기화한 후 다음 행 값을 추가한다.

4. 테이블정보 수집하기

1. 테이블을 찾는다.

학번	이름
101	김효관
102	이순신
103	김어진

```
import requests, bs4
import pandas as pd
```

#웹페이지 html 소스 가져오기

```
resp = requests.get("https://sparkkorea.com/테스트/")
resp.encoding='utf-8'
html = resp.text
bs = bs4.BeautifulSoup(html, 'html.parser')
```

```
tabletag = bs.find("table", {"id":"test_table"})
tabletag
```

```
<table class="type07" id="test_table">
<thead>
<tr>
<th scope="cols">학번</th>
<th scope="cols">이름</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>김 효관</td>
</tr>
<tr>
<td>102</td>
<td>이순신</td>
</tr>
<tr>
<td>103</td>
<td>김 어진</td>
</tr>
</tbody>
</table>
```

4. 테이블 정보 수집하기

2. tbody의 tr 태그를 모두 찾는다.

학번	이름
101	김효관
102	이순신
103	김어진

```
rows = tabletag.find("tbody").findAll("tr")
```

```
rows = tabletag.find("tbody").findAll("tr")  
rows
```

```
[<tr>  
  <td>101</td>  
  <td>김 효관</td>  
</tr>,  
<tr>  
  <td>102</td>  
  <td>이순신</td>  
</tr>,  
<tr>  
  <td>103</td>  
  <td>김어진</td>  
</tr>]
```

4. 테이블 정보 수집하기

3. 컬럼 값을 반복하면서 데이터를 수집한다.

학번	이름
101	김효관
102	이순신
103	김어진

행 전체를 저장할 리스트

rowList=[]

행별 컬럼값을 저장할 리스트

columnList = []

행을 반복하면서 td를 모두 찾는다

```
for i in range(0, len(rows)):
    columns = rows[i].findAll("td")
    columnsLen = len(columns)
    for j in range(0, columnsLen):
        columnList.append(columns[j].text)
```

```
rowList.append(columnList)
columnList=[]
```

행별 컬럼 데이터 수집 후 컬럼리스트 초기화

rowList

[['101', '김효관'], ['102', '이순신'], ['103', '김어진']]

4. 테이블정보 수집하기

4. 데이터 프레임으로 변환

학번	이름
101	김효관
102	이순신
103	김어진

```
pd.DataFrame(rowList,  
              columns = ["학번","이름"] )
```

한 행씩 담긴 반복 리스트를
데이터프레임 타입으로 변환

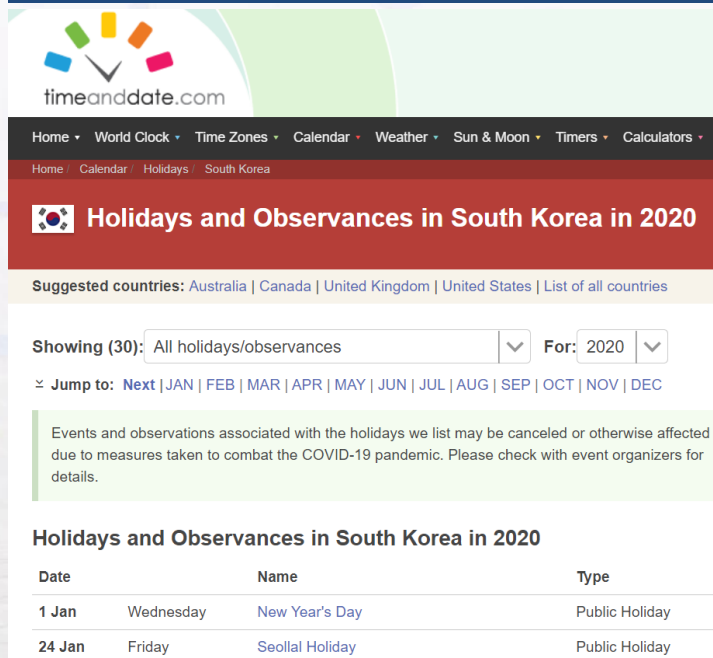
```
pd.DataFrame(rowList, columns =["학번","이름"])
```

	학번	이름
0	101	김효관
1	102	이순신
2	103	김어진

컬럼명 크롤링은 `thead` 활용
실습!!

[실습풀이] 태그정보 수집응용 실습

<https://www.timeanddate.com/holidays/south-korea/>



The screenshot shows the 'timeanddate.com' website. The main heading is 'Holidays and Observances in South Korea in 2020'. Below this, there's a section for 'Showing (30):' with a dropdown menu set to 'All holidays/observances' and a 'For:' dropdown set to '2020'. A note mentions that events may be affected by the COVID-19 pandemic. At the bottom, a table lists the holidays for 2020.

Date	Name	Type
1 Jan	Wednesday New Year's Day	Public Holiday
24 Jan	Friday Seollal Holiday	Public Holiday

	date	weekinfo	holiday_name	type
0	1월 1일	수요일	New Year's Day	Public Holiday
1	1월 24일	금요일	Seollal Holiday	Public Holiday
2	1월 25일	토요일	Seollal	Public Holiday
3	1월 27일	월요일	Seollal Holiday	Public Holiday

힌트 #1: tr을 패턴을 보고 스킵할 부분이 존재함
힌트 #2: iter 시 tr내 td 외 th 가 존재함

4. 핵심정리 및 Q&A

기억합시다

1

기초문법 라이브러리/변수 선언, 조건/반복문, 함수화 하는 방법을 기억한다

2

BeautifulSoup를 활용한 웹 크롤링 방법을 기억한다.

감사합니다.