3. 정적 크롤링

contents

- **BeautifulSoup 라이브러리**
- **BeautifulSoup 라이브러리 활용**
- **웹사이트 크롤링 실습**

1. BeautifulSoup 라이브러리

- **BeautifulSoup**

BeautifulSoup	▪ HTML 및 XML 파일에서 데이터를 추출하기 위한 파이썬 라이브러리

  - 파이썬 기본 라이브러리가 아니므로 별도 설치 필요
  - 설치 : `pip install beautifulsoup4`
  - Anaconda는 BeautifulSoup 패키지가 Site-packages로 설치되어 있음
  - HTML 및 XML 파일의 내용을 읽을 때 다음 파서(Parser) 이용

html.parser	lxml
lxml-xml	html5lib

1. BeautifulSoup 라이브러리

- **BeautifulSoup**
  - 파서 라이브러리(Parser Library)비교

파서	구현 방법	특징
Python's html.parser	BeautifulSoup (markup, "html.parser")	• 추가 설치 필요 없음 • 적당한 속도
lxml's HTML parser	BeautifulSoup (markup, "lxml")	• 추가 설치 필요 **없음** • 속도가 빠름
lxml's XML parser	BeautifulSoup (markup, "lxml-xml") BeautifulSoup (markup, "xml")	• 속도가 빠름 **없음** • 추가 설치 필요
html5lib	BeautifulSoup (markup, "html5lib")	• 속도가 느림 • 추가 설치 필요 • 웹 브라우저와 동일한 방식으로 페이지의 파싱 지원

```
! pip install beautifulsoup4 html5lib
```

1. BeautifulSoup 라이브러리

- **HTML 파싱**

1. BeautifulSoup의 메인 패키지인 bs 패키지에서 BeautifulSoup() 함수 임포트
2. 파싱할 HTML 문서와 파싱에 사용할 파서(구문 분석기)를 지정하여 호출
   ➡ bs4.BeautifulSoup 객체 리턴
3. HTML 문서에 대한 파싱이 끝나면 트리 구조 형식으로 DOM 객체 생성
   ➡ bs4.BeautifulSoup 객체를 통해 접근 가능

```
from bs4 import BeautifulSoup
bs = BeautifulSoup(html_doc, 'html.parser')
bs = BeautifulSoup(html_doc, 'lxml')
bs = BeautifulSoup(html_doc, 'lxml-xml')
bs = BeautifulSoup(html_doc, 'html5lib')
```

or bs=BeautifulSoup(html_doc,'xml')

2. BeautifulSoup 라이브러리 활용

- **bs4.BeautifulSoup 객체의 태그 접근 방법**
  - HTML 문서를 파싱하고 bs4.BeautifulSoup 객체 생성

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

  - <html>, <head> 태그와 <body> 태그는 제외하고 접근하려는 태그에 계층구조를 적용
  - 태그명을. 연산자와 함께 사용

```
bs.태그명
bs.태그명.태그명
bs.태그명.태그명.태그명
bs.태그명.태그명.태그명.태그명
```

2. BeautifulSoup 라이브러리 활용

- **태그 속성 정보 추출**
  - bs4.element.Tag객체의 주요 속성과 메서드

태그명 추출

bs.태그명.name

속성 추출

bs.태그명['속성명']
bs.태그명.attrs

콘텐츠 추출

bs.태그명.string
bs.태그명.text
bs.태그명.contents
bs.태그명.strings
bs.태그명.get_text()

bs.태그명.string.strip()
bs.태그명.text.strip()
bs.태그명.stripped_strings
bs.태그명.get_text(strip=True)

2. BeautifulSoup 라이브러리 응용

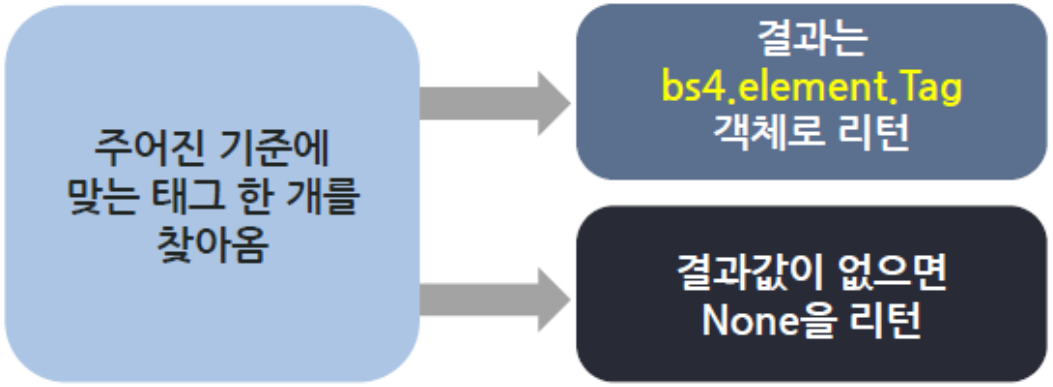
- **bs4.BeautifulSoup 객체의 주요 메서드**
  - HTML 문서에 대한 파싱이 끝나고 생성된 트리구조 형식의 DOM 객체
    - bs4.BeautifulSoup 객체의속성으로접근가능
  - 다음에 제시된 메서드로도 가능

태그 찾기 기능의 주요 메서드

- find_all()
- find()
- select()

2. BeautifulSoup 라이브러리 응용

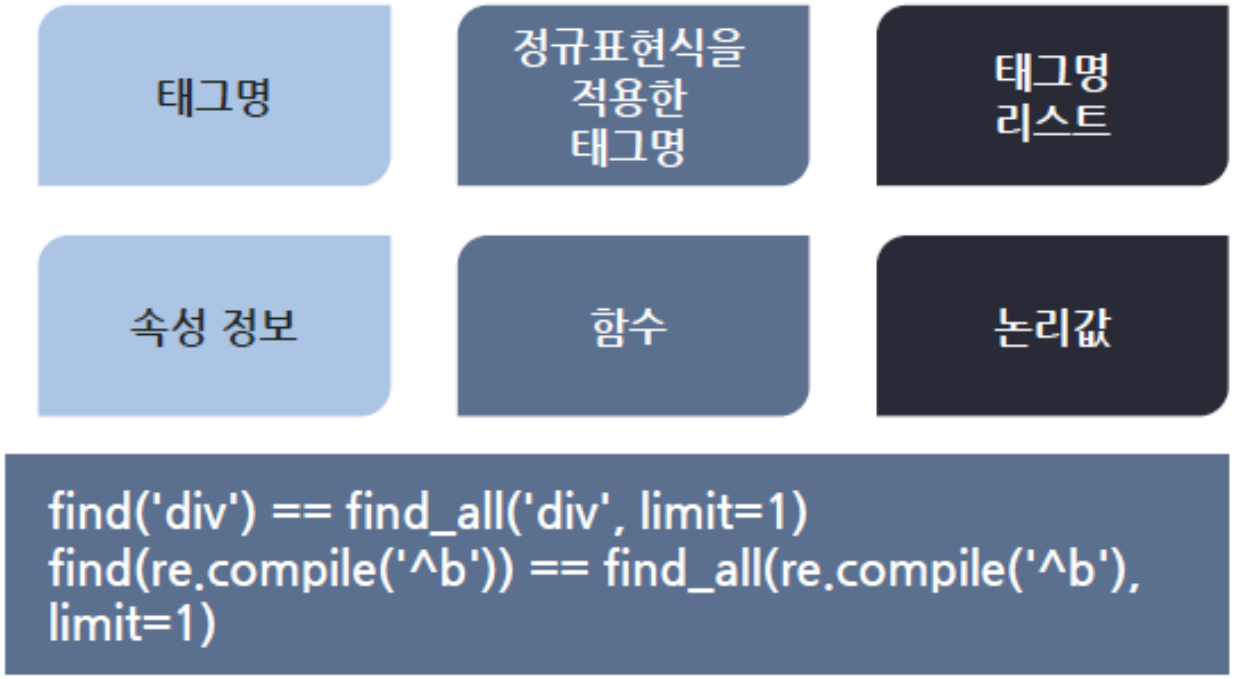
- 메서드를 사용한 웹페이지 파싱 : bs.find()




```
find(name=None, attrs={}, recursive=True, text=None, **kwargs)
```

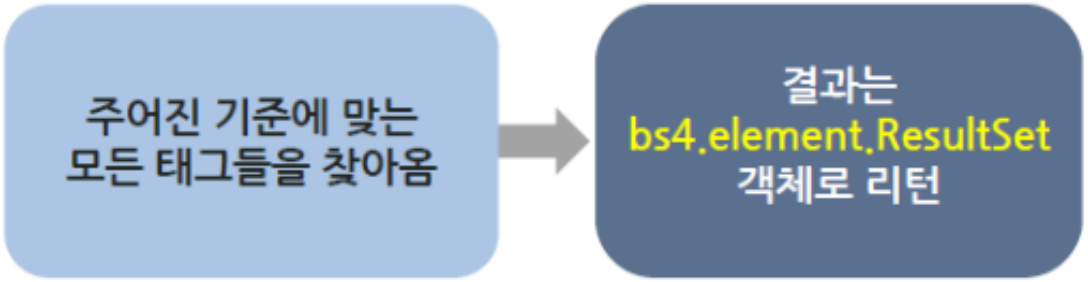
2. BeautifulSoup 라이브러리 응용

- **메서드를 사용한 웹페이지 파싱 : bs.find()**
  - 기준 설정



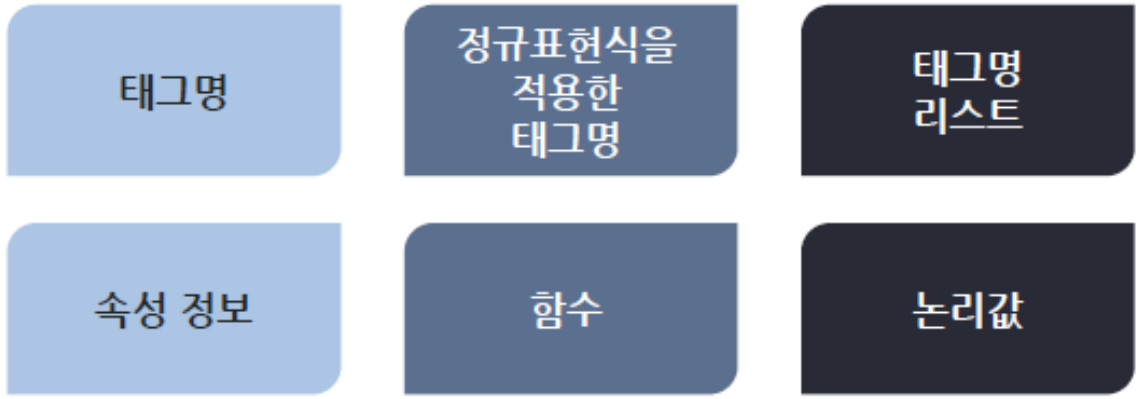
2. BeautifulSoup 라이브러리 응용

- **메서드를 사용한 웹페이지 파싱 : bs.find_all()**




```
find_all(name=None, attrs={}, recursive=True, text=None, limit=None, **kwargs)
```

- 기준 설정



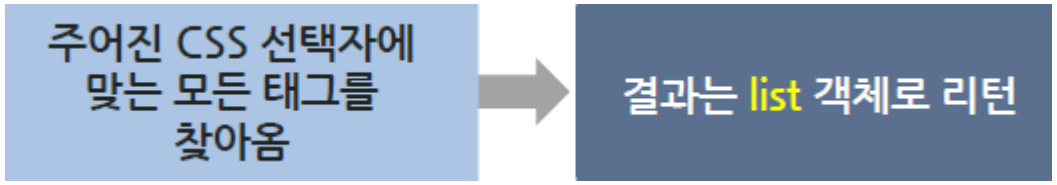
2. BeautifulSoup 라이브러리 응용

- **메서드를 사용한 웹페이지 파싱 : bs.find_all()**

```
find_all('div')
find_all(['p','img'])
find_all(True)
find_all(re.compile('^b'))
find_all(id='link2')
find_all(id=re.compile("para$"))
find_all(id=True)
find_all('a', class_="sister")
find_all(src=re.compile("png$"), id='link1')
find_all(attrs={'src':re.compile('png$'), 'id':'link1'})
find_all(text='example')
find_all(text=re.compile('example'))
find_all(text=re.compile('^test'))
find_all(text=['example', 'test'])
find_all('a', text='python')
find_all('a', limit=2)
find_all('p', recursive=False)
```

2. BeautifulSoup 라이브러리 응용

- 메서드를 사용한 웹페이지 파싱 : bs.select(), bs.select_one()



select_one(selector, namespaces=None, limit=None, **kwargs) 결과가 element
select(selector, namespaces=None, limit=None, **kwargs) : 결과가 리스트

  - CSS 선택자를 적용한 호출

```
select('태그명')
select('.클래스명')
select('#아이디명')
select('태그명.클래스명))
```

2. BeautifulSoup 라이브러리 응용

- **메서드를 사용한 웹페이지 파싱 : bs.select()**

  - 자식 선택자 및 자손 선택자를 사용하면 HTML문서의 트리 구조를 적용하여 태그를 찾을 수 있음

```
select('상위태그명 > 자식태그명 > 손자태그명')
select('상위태그명.클래스명 > 자식태그명.클래스명')
select('상위태그명.클래스명   자손태그명')
select('상위태그명 > 자식태그명  자손태그명')
select('#아이디명 > 태그명.클래스명)
select('태그명[속성]')
select('태그명[속성=값]')
select('태그명[속성$=값]')
select('태그명[속성^=값]')
select('태그명:nth-of-type(3)')
```

실습1: find(), find_all()

- 기상청 육상 중기예보 사이트로부터 도시별 시간별, 날씨, 최저기온, 최고 기온을 가져온다.

```
url="http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp"
```

```
서울
2025-03-08 00:00 구름많음 1 9
2025-03-08 12:00 맑음 1 9
2025-03-09 00:00 맑음 1 12
2025-03-09 12:00 맑음 1 12
2025-03-10 00:00 맑음 4 15
2025-03-10 12:00 맑음 4 15
2025-03-11 00:00 흐림 5 13
2025-03-11 12:00 흐림 5 13
2025-03-12 00:00 흐림 5 14
2025-03-13 00:00 구름많음 5 14
2025-03-14 00:00 구름많음 4 13
인천
2025-03-08 00:00 구름많음 1 7
2025-03-08 12:00 맑음 1 7
2025-03-09 00:00 맑음 2 10
2025-03-09 12:00 맑음 2 10
2025-03-10 00:00 맑음 4 12
2025-03-10 12:00 맑음 4 12
2025-03-11 00:00 흐림 5 12
2025-03-11 12:00 흐림 5 12
```

실습2: select(), select_all()

- **yes24 페이지에서 검색어를 입력하여 검색된 책 제목, 저자, 출판사, 출판연월, 가격, 평점을 스크랩하여 csv파일과 데이터베이스로 저장한다.**
- **책의 이미지를 스크래핑하여 data/images폴더에 이미지 파일을 저장한다.**

	title	author	publish	price	grade
0	Do it! 점프 투 파이썬	박응용	이지스퍼블리싱	19800	9.8
1	밑바닥부터 시작하는 딥러닝 1	사이토 고키	한빛미디어	23400	9.2
2	혼자 만들면서 공부하는 파이썬	문현일	한빛미디어	23400	10.0
3	두근두근 파이썬	천인국	생능출판사	25000	5.0
4	혼자 공부하는 파이썬	윤인성	한빛미디어	19800	9.5
...	...	...	...	...	...
67	작심 3일 파이썬 Python	황덕창	스포트라잇북	16200	0.0
68	파이썬 Python을 이용한 통계학 원론	권세혁	교우사(오판근)	24000	0.0
69	PYTHON 파이썬만 잡아도	김병만	카오스북	25000	0.0
70	The Python - 파이썬 프로그래밍	허진경	BOOKK(부크크)	32000	0.0
71	Getting Start Python(파이썬)3rd	김영아	구민사	23400	0.0