

# 빅데이터 수집시스템 개발



## 다양한 웹 사이트의 웹 크롤링 실습

## 학습목표

- 실전 예제를 통해 HTML과 CSS만으로 구성된 웹 콘텐츠의 크롤링 및 스크래핑을 할 수 있다.
- 웹 크롤링 프로그램의 정보를 변경하여 웹 서버에 요청하는 방법을 익힐 수 있다.

## 학습내용

- 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제
- 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 1 크롤링할 URL과 화면

- 크롤링 URL

[https://movie.n\\*\\*\\*\\*.com/movie/point/af/list.nhn?page=1](https://movie.n****.com/movie/point/af/list.nhn?page=1)

- 스크래핑 내용

- 영화 제목
- 평점
- 댓글

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 1 크롤링할 URL과 화면

영화	네티즌 평점·140자평		
	개봉 전 평점		개봉 후 평점
	평점		140자평
			글쓴이 날짜
영화 홈			
상영작, 예정작			
영화랭킹			
예매			
평점, 리뷰			
▶네티즌 평점			
네티즌 리뷰			
다운로드			
	★★★★★ 10	<b>말할 수 있는 비밀</b> 처음봤을 땐 상륜의 입장에서 보게 되고, 두 번째부터는 샤오위의 입장에서 보게 된다.	ing*** **06.11
	★★★★★ 9		sona*** **06.11
		008 에바그린... 영화에 빠져드는 치명적인 매력이 아 직도 기억에 남는다. 결말이 너무 아쉬웠다.	

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 1 크롤링할 URL과 화면

총 10개

- |        |    |  |
|--------|----|--|
| ★★★★★★ | 10 | <b>말할 수 있는 비밀</b><br>처음봤을 땐 상륜의 입장에서 보게 되고,<br>두 번째부터는 ...     |
| ★★★★★  | 9  | <b>008</b><br>에바그린. 영화에 빠져드는 치명적인 매력<br>이 아직도 기억에 남는다.         |
| ★☆☆☆☆  | 1  | <b>독고하이</b><br>와. 투자자. 돈만 있으면 진심 나도 이정<br>도는 만들듯...            |
| ★★★★★  | 9  | <b>와이맨 : 다크 피크닉</b><br>전작에 비해 못한건 맞지만 별점을 짜게<br>주는 사람이 너무 많네.. |
| ★★★★★  | 9  | <b>여름왕국</b><br>포세이돈의 형상화가 너무 멋졌음                               |
| ★★★★★★ | 10 | <b>기생충</b><br>걱정을 해서 걱정이 없어지면 걱정이<br>없겠네.                      |
| ★☆☆☆☆  | 1  | <b>유랑</b><br>아주 유치한 영화입니다.                                     |
| ★★★★★★ | 10 | <b>기생충</b><br>너무 사실적인 영화. 서로 가진 것에 만족하<br>고 인정하며 그저 이 땅에서...   |
| ★☆☆☆☆  | 1  | <b>보이캡스</b><br>관람객 평점에 낚여서 봤는데 전개, 대사<br>다 너무 뻘하네요.            |
| ★★★★★  | 9  | <b>가서</b><br>이 영화 좀 피폐함...                                     |

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 2 스크래핑 설계

번호	감상평
16433038	<b>잡범</b> ★★★★★ 9 삼박자 좋다. 연기가 특히 좋다. 인기있는 배우가 나오지 않는다고 실망하지 마라. 연기 잘하는 배우들이 나온다. 처음보고 놀랐다가 두 번째 보고 소름 돋았다. 스릴러 좋아하는 사람들에게 추천한다. 재미지다.
16433037	<b>여름왕국</b> ★☆☆☆☆ 2 다양한 층을 겨냥하여 다양한 장르로 시도하다보니 킬링 포인트가 부족한 느낌...

평점 : td.title > div > em의 콘텐츠

영화 제목 : .movie의 콘텐츠

댓글 : td.title의 콘텐츠 중 7번째 자식 콘텐츠

```

<tr>
  <td class="ac_num">16433038</td>
  <td class="title">
    <a href="?st=mcode&sword=174834&target=after"
      class="movie_color_b">진범</a>
    <div class="list_netizen_score">
      <span class="st_off">...</span>
      <em>9</em>
    </div>
    <br>
    "삼박자 좋다. 연기가 특히 좋다. 인기있는 배우가 나오지 않는다고 실망하지 마라. 연기 잘하는 배우들이 나온다. 처음보고 놀랐다가 두 번째 보고 소름 돋았다. 스릴러 좋아하는 사람들에게 추천한다. 재미지다."
  </td>
</tr>

```

## 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

## 3 소스

```
#파일명 : exam5_1.py
import requests
from bs4 import BeautifulSoup
import re

req =
requests.get('http://movie.n****.com/movie/point/af/list.nhn?p
age=1')
html = req.text
soup = BeautifulSoup(html, 'html.parser')
titles = soup.select('.movie')
points = soup.select('td.title > div > em')
reviews = soup.select('td.title')

movie_title = []
movie_point = []
movie_review = []
for dom in titles:
    movie_title.append(dom.text)
for dom in points:
    movie_point.append(dom.text)
for dom in reviews:
    content = dom.contents[6]
    content=re.sub("신고", "", content)
    content=re.sub("[\W\n\Wt]", "", content)
    movie_review.append(content)
commentLength = len(movie_title)
for i in range(commentLength):
    print("영화 제목 : " + movie_title[i])
    print("평점 : " + movie_point[i])
    print("리뷰글 : " + movie_review[i])
    print("-----")
```

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 4 실행 결과 화면

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myvscode/unit5/exam5_1.py
```

영화 제목 : 완벽한 남자

평점 : 10

리뷰글 : 울다 웃다ㅋㅋㅋ 배우들 케미ㅋㅋㅋ 허준호도 왜 이렇게 멋져? 아 이 영화 진짜 너무 좋아~~~

기대 없이 봤다가 허리 세워서 스크린에 빠져 들어서 봄ㅋㅋㅋ 신랑이랑 또 보러 가야지!!!

-----

영화 제목 : 호랑이

평점 : 4

리뷰글 : 금수저와 흙수저의 싸움, 흙수저는 자신이 죽을 각오로 다 버리고 싸우는데 금수저는 아빠 버프와 지인 버프까지 더해져 폼나는 불주먹으로 미친한 흙수저를 쓰러뜨린다~ 심지어 아빠는 환하게 웃으며 이런것쯤 아무 것도 아니라는듯 도움을~ㅋㅋㅋ

-----

영화 제목 : 얼굴없는 부하

평점 : 2

리뷰글 : 얼굴없는 부하를 위해 다른 부하를 대신 죽이고 바다에 빠지는 장면 거기서 빵터짐..



# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 5 여러 페이지 크롤링

- 크롤링 URL

[https://movie.n\\*\\*\\*\\*.com/movie/point/af/list.nhn?page=1](https://movie.n****.com/movie/point/af/list.nhn?page=1)

영화 홈	★★★★★★ 10	알라딘
상영작, 예정작		지니가 최고였어.
영화랭킹	★★★★★ 9	요술
예매		이것도 영화라고.
평점, 리뷰	★★★★★★ 10	미녀
▶ 네티즌 평점		와우 신선해

1 2 3 4 5 6 7 8 9 10 >

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 5 여러 페이지 크롤링

- 크롤링 URL

[https://movie.n\\*\\*\\*\\*.com/movie/point/af/list.nhn?page=n](https://movie.n****.com/movie/point/af/list.nhn?page=n)

[https://movie.n\\*\\*\\*\\*.com/movie/point/af/list.nhn?page=2](https://movie.n****.com/movie/point/af/list.nhn?page=2)

영화 홈	★★★★★★ 10	알라딘
상영작, 예정작	★★★★★ 9	요술
영화랭킹	★★★★★★ 10	미녀
예매		
평점, 리뷰		
▶ 네티즌 평점	1 2 3 4 5 6 7 8 9 10 >	

[https://movie.n\\*\\*\\*\\*.com/movie/point/af/list.nhn?page=3](https://movie.n****.com/movie/point/af/list.nhn?page=3)

영화 홈	★★★★★★ 10	알라딘
상영작, 예정작	★★★★★ 9	요술
영화랭킹	★★★★★★ 10	미녀
예매		
평점, 리뷰		
▶ 네티즌 평점	1 2 3 4 5 6 7 8 9 10 >	

## 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

## 6 여러 페이지 크롤링 소스

```
#파일명 : exam5_2.py
import requests
from bs4 import BeautifulSoup
import re
for n in range(1,31):
    req =
requests.get('http://movie.n****.com/movie/point/af/list.nhn?p
age='+str(n))
    html = req.text
    soup = BeautifulSoup(html, 'html.parser')
    titles = soup.select('.movie')
    points = soup.select('td.title > div > em')
    reviews = soup.select('td.title')
    movie_title = []
    movie_point = []
    movie_review = []
for dom in titles:
    movie_title.append(dom.text)
for dom in points:
    movie_point.append(dom.text)
for dom in reviews:
    content = dom.contents[4]
    content=re.sub("신고", "", content)
    content=re.sub("[\n\t]", "", content)
    movie_review.append(content)

commentLength = len(movie_title)
for i in range(commentLength):
    print(movie_point[i] +
"\t"+movie_title[i)+"\t"+movie_review[i])
print("-----")
```

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 1 영화 웹 페이지에서 현재 상영작에 대한 네티즌 평점과 댓글

### 7 여러 페이지 크롤링 실행 결과 화면

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/
myvscode/unit5/exam5_2.py
```

2 여름왕국 다양한 층을 겨냥하여 다양한 장르로 시도때도 없이 틀어대는 노래들과 너무  
다양하게 시도하다보니 뭔가 킬링포인트가 없고 보다가 자꾸 산만해짐... 올라프야 고생했어!!  
너 덕분에 그나마 웃고간다~

8 여름왕국 포세이돈의 형상화가 너무 멋졌음



10 연희에게 정말 재미있게 잘 봤습니다!!! 다만 아쉬운 점은 상영관이 많이 없는거? 그  
외 모두 만족하고 못 보신 분들 추천드려요!!

10 알바 많은 생각을 하게 하는 영화네요 ㅠㅠ 최고!

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 2 서점 웹 페이지에서 파이썬 관련 서적 정보

### 1 크롤링할 URL과 화면

- 크롤링 URL

```
http://www.y***4.com/SearchCorner/Search?domain=BOOK&query=python
```

- 스크래핑 내용

- 도서 제목
- 도서 제목에 지정된 링크

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 2 서점 웹 페이지에서 파이썬 관련 서적 정보

### 1 크롤링할 URL과 화면

국내도서 : "Python" 검색 결과 1-20 / 381건

인기도	정확도	신상품	최저가	최고가	평점순	리뷰순
1						
	<p>[도서] 할 수 있다! 레벨 UP 파이썬</p> <p>홍길동 저   OO출판   2050년 03월</p> <p>18,800원 → <b>16,920원</b>(10% 할인)      포인트 940원(5%)</p> <p>도착 예상일 : 지금 주문하면 오늘 도착 예정</p> <p>#파이썬   #컴공   #개발자   #프로그래밍</p>					
2						
	<p>[도서] 밑바닥부터 시작하는 Python</p> <p>김철수 저   XX미디어   2055년 01월</p>					

20개

```

Elements  Console  Sources  >>  ⋮  >
<td class= goods_img >...</td>
<td class="goods_infogr">
  <p class="goods_name
goods_icon">
    "
                                [도서]
                                "
    <a href="/Product/Goods/
24567417?scode=032&OzSrank=1">
      <strong>
파이썬</strong> == $0
    </a>
    <span class="goods_sname">
</span>
  
```

# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 2 서점 웹 페이지에서 파이썬 관련 서적 정보

### 3 소스

```
#파일명 : exam5_3.py
import requests
from bs4 import BeautifulSoup
title = []
link = []
urlstr =
'http://www.y***4.com/SearchCorner/Search?domain=BOOK&q
uery=
python'
r = requests.get(urlstr)
#r.encoding = "utf-8"
bs = BeautifulSoup(r.text, 'html.parser')
titleList = bs.select('p.goods_name.goods_icon > a > strong')
linkList = bs.select('p.goods_name.goods_icon > a')
for titleDom in titleList:
    title.append(titleDom.string)
for linkDom in linkList:
    link.append(linkDom["href"])

print("-- 도서 제목 --")
print(title)
print("-- 도서 링크 URL --")
print(link)
```



# 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 2 서점 웹 페이지에서 파이썬 관련 서적 정보

### 4 실행 결과 화면

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/
myvscode/unit5/exam5_3.py
-- 도서 제목 --
[ ' 독학으로 배우는 파이썬 ', ' 할 수 있다! 레벨 UP 파이썬 ', ' 밑바닥부터 시
작하는 Python ', ' it is 파이썬 ', ' 딥러닝 초보 탈출! ', ' 딥러닝 교과서 -
차근차근 밟아가는 파이썬 ', ' 머신러닝 in 파이썬 ', ' 데이터 분석을 위한 파
이썬 ', ' 파이썬 한 그릇 똑딱 ', ' 실전 파이썬 웹 프로그래밍 ', ' 컴퓨터 알고
리즘 with 파이썬 ', ' 라즈베리 파이 4를 통한 IoT ', ' 데이터 분석을 위한 파
이썬 라이브러리 탐구 ', ' 영상처리 in 파이썬 ', ' 파이썬으로 통계학 시작! ',
' 너와 나의 파이썬 ', ' 파이썬 한 권 완벽 가이드 ', ' 파이썬 + 딥러닝 뽀개기
', ' 응용 텍스트 분석을 위한 파이썬 ', ' 파이썬 실전 Guide ' ]
-- 도서 링크 URL --
[ '/Product/Goods/74269975?scode=032&0zSrank=1', '/campaign/00_corp/2019bo
y/bookAward_book.aspx', '/Product/Goods/74419916?scode=032&0zSrank=2', '/
Product/Goods/34970929?scode=032&0zSrank=3', '/Product/Goods/79672557?sco
de=032&0zSrank=4', '/Product/Goods/73270768?scode=032&0zSrank=5', '/Produ
```



## 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제

## 2 서점 웹 페이지에서 파이썬 관련 서적 정보

## 5 주식 해제시 실행 결과 화면

```

PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myv
scode/unit5/exam5_3.py
-- 도서 제목 --
['?', 'UP?', 'UP?+?', 'with?', '2', 'â?', '']
1
2
3 <!DOCTYPE html>
4 <html lang="ko">
5
6 <head>
7 <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
8
9 <meta http-equiv="Content-Type" content="text/html; charset=euc-kr" />
10
11 <meta name="viewport" content="width=1170" />
12 <title>YES24 | 대한민국 대표 인터넷서점</title>
13
14 <meta name="title" content="YES24 - 대한민국 대표 인터넷서점" />
15 <meta name="description" content="YES24는 대한민국 1위 인터넷 온라인 서점 입니다.
    양한 문화 콘텐츠 및 서비스를 제공합니다." />

```

# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 1 출판 네트워크 웹 페이지의 회원 마일리지와 이코인 정보

### 1 크롤링할 URL과 화면

1 회원이 부여받은 마일리지, 이코인 정보를 웹 크롤링을 통해서 추출

- 크롤링 URL

`http://www.h****t.co.kr/myh**bit/myh**bit.htm`

- 스크래핑 내용

- 회원 마일리지
- 회원 이코인

# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 1 출판 네트워크 웹 페이지의 회원 마일리지와 이코인 정보

### 1 크롤링할 URL과 화면

### 2 로그인이 필요한 서비스라는 경고창 출력




# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 1 출판 네트워크 웹 페이지의 회원 마일리지와 이코인 정보

### 1 크롤링할 URL과 화면

### 3 마일리지와 이코인 정보 화면 출력

HOME 미디어 아카데미 비즈 라이프 에듀					
OO출판 네트워크			BRAND	Channel.O	
OO멤버십	마일리지/OO이코인	위시리스트	장바구니	구매이력(주문조회)	My 쿠폰
 <p>Family</p> <p>(홍길동)님의 회원 등급은 일반(준회원)입니다.</p>		<div>마일리지 0점</div> <div>OO이코인 0원</div>		<div>최근 구매이력</div> <div>주문일자</div> <div>2050.11.08    You Don't</div> <div>2050.08.13    처음 시작하</div>	

# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 1 출판 네트워크 웹 페이지의 회원 마일리지와 이코인 정보

### 2 스크래핑 설계

HOME

미디어

아카데미

비즈

라이프

예뻐

OO출판 네트워크

BRAND

Channel.O

OO멤버십

마일리지/OO이코인

위시리스트

장바구니

구매이력(주문조회)

My 쿠폰

Family

(홍길동)님의

회원 등급은 일반(준회원)입니다.

마일리지

0점

OO이코인

0원

최근 구매이력

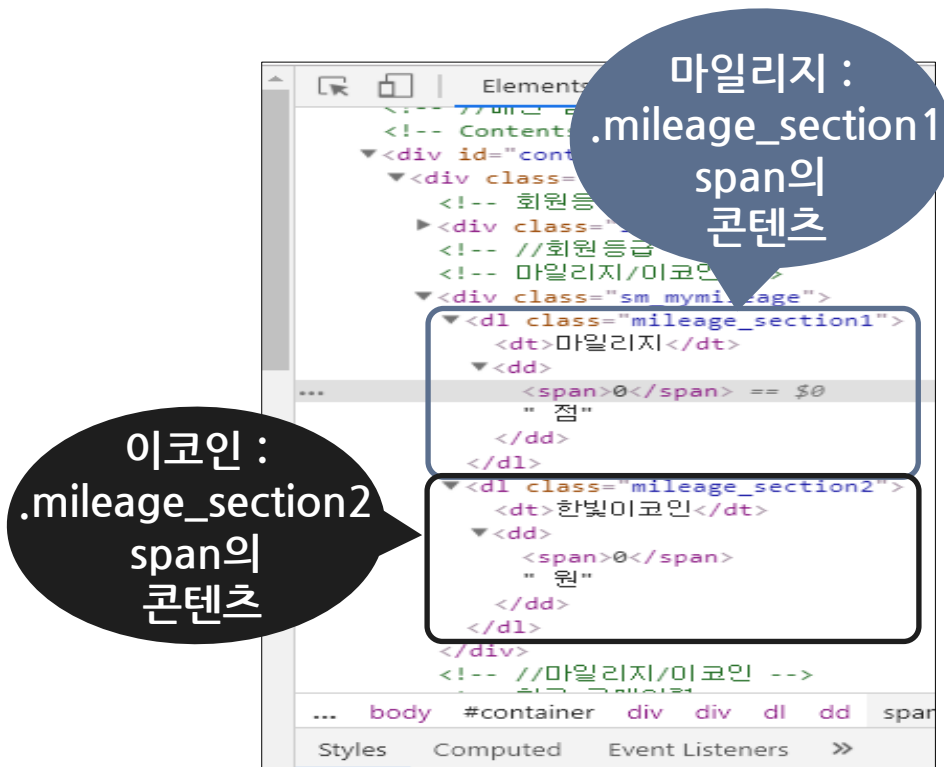
주문일자

2050.11.08

You Don't

2050.08.13

처음 시작하



# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 1 출판 네트워크 웹 페이지의 회원 마일리지와 이코인 정보

### 3 소스

```
#파일명 : exam5_4.py
import requests
from bs4 import BeautifulSoup
session = requests.session()
login_info = {
    "m_id": "계정",
    "m_passwd": "패스워드"
}

url_login = "http://www.h****t.co.kr/member/login_proc.php"
res = session.post(url_login, data=login_info)
res.raise_for_status()
url_mypage = "http://www.h****t.co.kr/myhanbit/myhanbit.html"
res = session.get(url_mypage)
res.raise_for_status()

soup = BeautifulSoup(res.text, "html.parser")
mileage = soup.select_one(".mileage_section1 span").get_text()
ecoin = soup.select_one(".mileage_section2 span").get_text()
print("마일리지:" + mileage)
print("이코인:" + ecoin)
```



# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 1 출판 네트워크 웹 페이지의 회원 마일리지와 이코인 정보

### 4 실행 결과 화면

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myv
scode/unit5/exam5_4.py
<class 'requests.models.Response'>
마일리지:0
이코인:0
```

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myv
scode/unit5/exam5_4.py
Traceback (most recent call last):
  File "c:/example/myvscode/unit5/exam5_4.py", line 21, in <module>
    mileage = soup.select_one(".mileage_section1 span").get_text()
AttributeError: 'NoneType' object has no attribute 'get_text'
```

## 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

### 2 기상청 웹 페이지의 기상청 육상 중기예보

#### 1 크롤링할 URL과 화면

- 크롤링 URL

`http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108`

- 스크래핑 내용

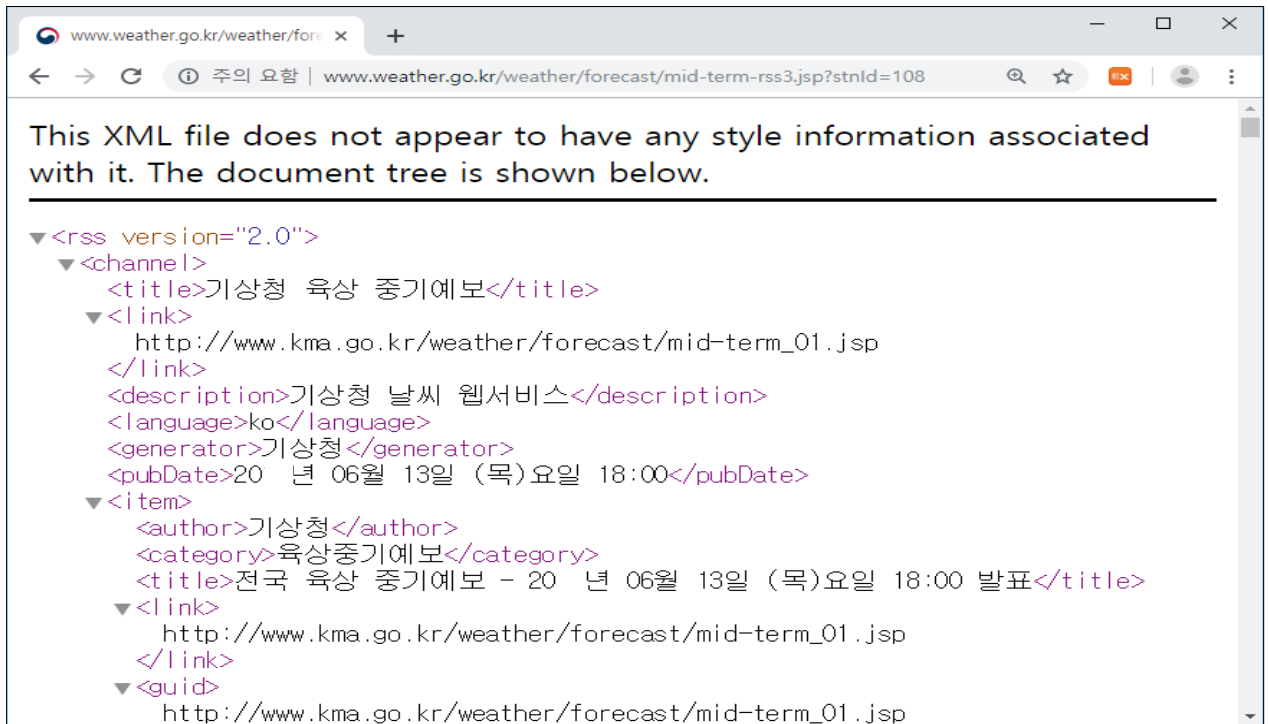
- 도시명
- 최저온도
- 최고온도



# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 2 기상청 웹 페이지의 기상청 육상 중기예보

### 1 크롤링할 URL과 화면



## 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 2 기상청 웹 페이지의 기상청 육상 중기예보

## 1 크롤링할 URL과 화면

The screenshot displays the XML structure of the weather forecast page. The left pane shows the root XML structure, and the right pane shows the detailed data for a specific location (Seoul).

**Left Pane (XML Structure):**

```

<rss version="2.0">
  <channel>
    <title>기상청 육상 중기예보</title>
    <link>http://www.kma.go.kr/weather/forecast/</link>
    <description>기상청 날씨 웹서비스</description>
    <language>ko</language>
    <generator>기상청</generator>
    <pubDate>20년 06월 13일 (목)요일 18:00</pubDate>
    <item>
      <author>기상청</author>
      <category>육상중기예보</category>
      <title>전국 육상 중기예보 - 20년 06월 13일</title>
      <link>http://www.kma.go.kr/weather/forecast/</link>
      <guid>http://www.kma.go.kr/weather/forecast/</guid>
    </item>
  </channel>
</rss>

```

**Right Pane (Detailed Data for Seoul):**

```

<location wl_ver="3">
  <province>서울·인천·경기</province>
  <city>서울</city>
  <data>
    <mode>A02</mode>
    <tmEf>20-06-16 00:00</tmEf>
    <wf>맑음</wf>
    <tmn>18</tmn>
    <tmx>28</tmx>
    <reliability>보통</reliability>
  </data>
  <data>
    <mode>A02</mode>
    <tmEf>20-06-16 00:00</tmEf>
    <wf>맑음</wf>
    <tmn>18</tmn>
    <tmx>28</tmx>
    <reliability>보통</reliability>
  </data>
  <data>
    <mode>A02</mode>
    <tmEf>20-06-17 00:00</tmEf>
    <wf>맑음</wf>
  </data>
</location>

```

**Callouts:**

- 최저온도 : tmn 태그의 콘텐츠** (Minimum temperature: tmn tag content) - Points to the `<tmn>18</tmn>` tag.
- 도시명 : city 태그의 콘텐츠** (City name: city tag content) - Points to the `<city>서울</city>` tag.
- 최고온도 : tmx 태그의 콘텐츠** (Maximum temperature: tmx tag content) - Points to the `<tmx>28</tmx>` tag.

# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 2 기상청 웹 페이지의 기상청 육상 중기예보

### 3 소스

```
#파일명 : exam5_5.py
from bs4 import BeautifulSoup
import urllib.request as req
import io
url = "http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108"
savename = "C:/Temp/forecast.xml"
req.urlretrieve(url, savename)
xml = open(savename, "r", encoding="utf-8").read()
soup = BeautifulSoup(xml, 'html.parser')
info = {}
for location in soup.find_all("location"):
    loc = location.find('city').string
    min_w = location.find_all('tmn')
    max_w = location.find_all('tmx')
    weather = [a.string+"~"+b.string for a, b in zip(min_w, max_w)]

    if not (loc in info):
        info[loc] = []
    for data in weather:
        info[loc].append(data)
print(info)

with open('C:/Temp/forecast.txt', "wt", encoding="utf-8") as f:
    for loc in sorted(info.keys()):
        f.write(str(loc)+'\n')
        for name in info[loc]:
            f.write('Wt'+str(name)+'\n')
```



# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 2 기상청 웹 페이지의 기상청 육상 중기예보

### 4 실행 결과 화면

```
PS C:\example\myvscod> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myvscod/unit5/exam5_5.py
{'서울': ['1~9', '1~9', '-1~9', '-1~9', '0~8', '0~8', '2~8', '2~8', '-1~6', '-1~6', '-3~5', '-3~5', '-4~5'], '인천': ['2~8', '2~8', '1~7', '1~7', '1~8', '1~8', '2~8', '2~8', '0~5', '0~5', '-2~4', '-2~4', '-1~5'], '수원': ['-1~10', '-1~10', '-1~9', '-1~9', '0~9', '0~9', '1~8', '1~8', '-1~5', '-1~5', '-2~5', '-3~4', '-4~5'], '파주': ['-2~8', '-2~8', '-4~8', '-4~8', '-4~8', '-4~8', '-2~7', '-2~7', '-5~5', '-5~5', '-7~4', '-8~4', '-7~5'], '이천': ['-2~9', '-2~9', '-3~9', '-3~9', '-3~9', '-3~9', '0~8', '0~8', '-2~6', '-2~6', '-5~5', '-5~5', '-6~4'], '평택': ['-1~10', '-1~10', '-1~10', '-1~10', '0~10', '0~10', '2~9', '2~9', '-1~6', '-1~6', '-3~6', '-2~5', '-4~6'], '춘천': ['0~8', '0~8', '-2~8', '-2~8', '-2~8', '-2~8', '1~7', '1~7', '-1~6', '-1~6', '-4~4', '-5~3', '-5~3'], '원주': ['0~9', '0~9', '-1~8', '-1~8', '-1~9', '-1~9', '2~8', '2~8', '0~7', '0~7', '-4~4', '-4~4', '-4~4'], '강릉': ['4~7', '4~7', '2~10', '2~10', '3~13', '3~13', '6~11', '6~11', '5~10', '5~10', '3~8', '2~9', '1~9'], '대전': ['0~11', '0~11', '-1~10', '-1~10', '0~11', '0~11', '4~9', '4~9']}
```

# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 2 기상청 웹 페이지의 기상청 육상 중기예보

### 4 실행 결과 화면

```

forecast.xml - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0">
<channel>
<title>기상청 육상 중기예보</title>
<link>http://www.kma.go.kr/weather/forecast/mid-term_01.jsp</link>
<description>기상청 날씨 웹서비스</description>
<language>ko</language>
<generator>기상청</generator>
<pubDate>20 年 11월 25일 (월)요일 06:00</pubDate>
<item>
<author>기상청</author>
<category>육상중기예보</category>
<title>전국 육상 중기예보 - 20 年 11월 25일 (월)요일 06:00 발표</title>
<link>http://www.kma.go.kr/weather/forecast/mid-term_01.jsp</link>
<guid>http://www.kma.go.kr/weather/forecast/mid-term_01.jsp</guid>
<description>
<header>
<title>전국 육상중기예보</title>
<tm>20 11250600</tm>
<wf><![CDATA[동풍의 영향으로 28일에 강원영동에 비 또는 눈이 오겠고, 기압골의 영향<
</header>
<body>

<location wl_ver="3">

```

```

forecast.txt - ...
파일(F) 편집(E) 서식(O) 보기(V)
도움말
서울
1~9
1~9
-1~9
-1~9
0~8
0~8
2~8
2~8
-1~6
-1~6
-3~5
-3~5
-4~5
세종
-1~11
-1~11
-2~10
-2~10
-2~11
-2~11
2~9
2~9
Windows (CRLF) UTF-8

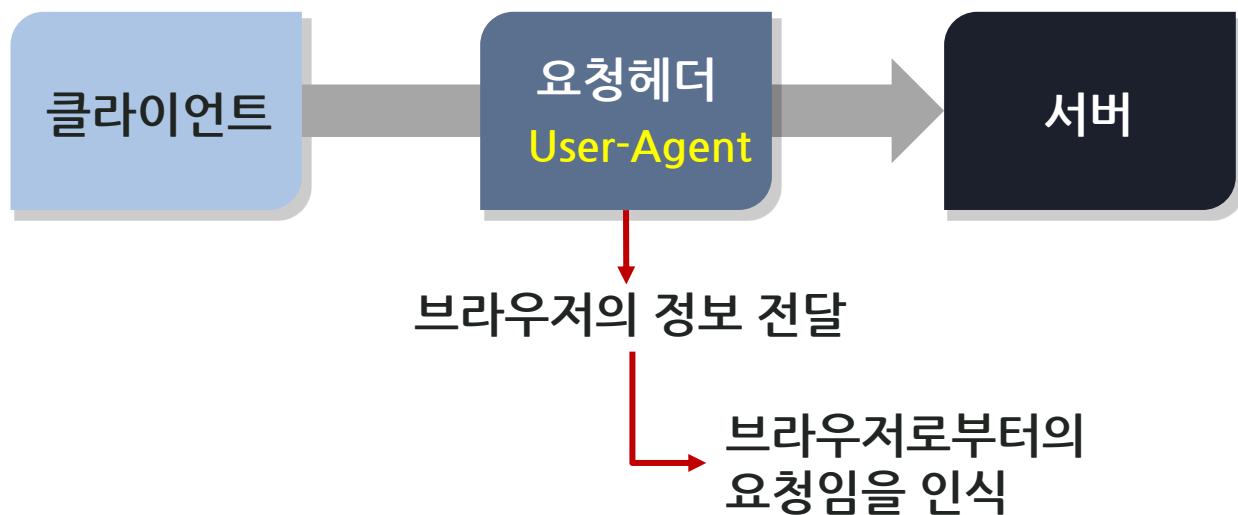
```

# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 3 클라이언트 정보를 변경하여 웹 크롤링

### 1 크롤링할 URL과 화면

- HTTP 통신



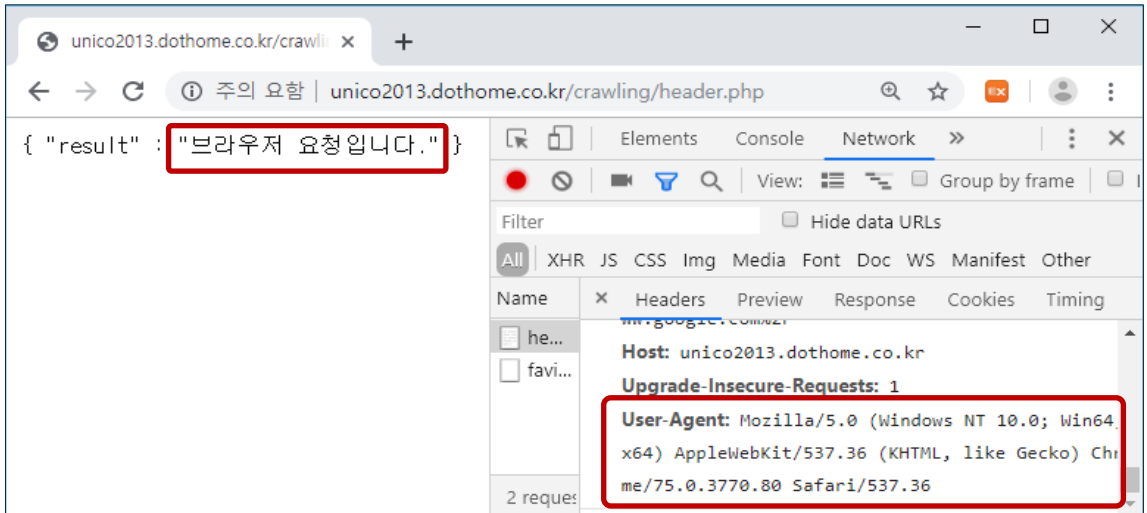
# 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 3 클라이언트 정보를 변경하여 웹 크롤링

### 1 크롤링할 URL과 화면

- 크롤링 URL

<http://unico2013.dothome.co.kr/crawling/header.php>



## 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

## 3 클라이언트 정보를 변경하여 웹 크롤링

## 1 크롤링할 URL과 화면

- 웹 크롤링하려는 사이트에서 브라우저로부터의 요청인지를 체크하는 경우



```
hdr = { ' User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) '+
```

```
      'AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.80 Safari/537.36'}
```

```
req = urllib.request.Request('요청 URL', headers = hdr)
```



## 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

### 3 클라이언트 정보를 변경하여 웹 크롤링

#### 2 소스

```
파일명 : exam5_6.py
import json
import urllib.request

#User-Agent를 조작하는 경우
hdr = {'User-agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) '+'
      'AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.80 Safari/537.36'}

req = urllib.request.Request('http://unico2013.dothome.co.kr/crawling/header.php',
                              headers = hdr)
#req = urllib.request.Request('http://unico2013.dothome.co.kr/crawling/header.php')
data = urllib.request.urlopen(req).read()
page = data.decode('utf-8', 'ignore')
res_content = json.loads(data)
print(res_content["result"])
```

## 출판 네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

### 3 클라이언트 정보를 변경하여 웹 크롤링

#### 3 실행 결과 화면

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myv  
scode/unit5/exam5_6.py  
브라우저 요청입니다.
```

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myv  
scode/unit5/exam5_6.py  
브라우저 요청이 아닙니다.
```

## 학습정리

### 1. 영화 및 서점 웹 페이지 크롤링과 스크래핑 예제



- 웹 페이지를 크롤링할 때 제일 먼저 체크해야 하는 것은 URL 문자열의 구조임
- 스크래핑하려는 웹 페이지의 소스 내용을 체크하여 어떤 태그의 내용 및 속성의 값을 추출할 것인지 판단
  - class 속성의 값 또는 id 속성의 값을 찾음
  - 부모와 자손 관계를 파악

## 학습정리

### 2. 출판네트워크, 기상청 및 클라이언트 정보 변경 크롤링과 스크래핑 예제

- CSS 선택자 중 클래스 선택자, 아이디 선택자, 자손 선택자, 속성 선택자를 주로 사용
- POST 방식 요청은 추가로 전달해야 하는 요청 파라미터에 대한 정보를 파악해야 하는데 이 때에는 크롬의 개발자 도구에서 Form Data 항목을 읽어봄
- 웹 서버에 요청을 보낼 때 클라이언트의 정보는 User-Agent라는 요청 헤더를 통해 전달