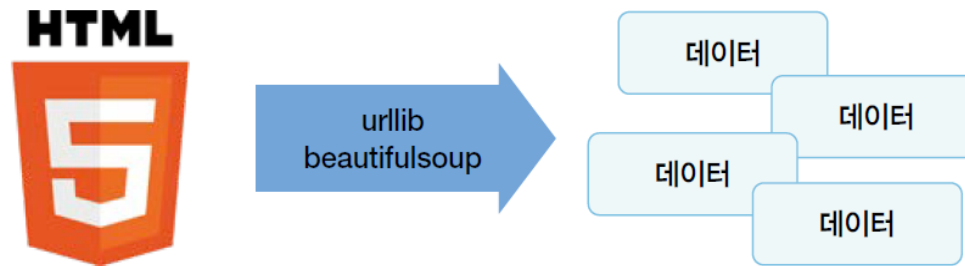


## Section 01 이 장에서 만들 프로그램

- [프로그램 1] HTML 데이터 추출
  - HTML에 표현된 여러 건의 데이터를 추출하는 프로그램



## Section 02 웹 크롤링 기본

### ■ 웹 크롤링과 웹 크롤러

- 인터넷에 공개된 데이터를 가져와 필요한 형식으로 변환하는 것이 데이터를 수집하는 가장 빠른 방법
- 크롤링(Crawling): 웹상에 공개된 내용에서 데이터를 추출하는 것
- 크롤러(Crawler): 웹 크롤링 작업을 하는 프로그램

## Section 02 웹 크롤링 기본

### ■ HTML 문법

- HTML: HyperText Markup Language의 약자  
웹 페이지를 만들기 위한 대표적인 마크업 언어
- HTML을 문법 중 부등호(<>) 안에 들어 있는 구문을 태그(Tag)라고 부름

### ■ HTML 태그의 특징

- HTML 파일의 확장자는 \*.htm 또는 \*.html
- HTML 파일은 텍스트 파일이므로 메모장 등에서 작성하면 됨  
단, 웹 브라우저에서 한글이 깨져 보일 수 있으므로 인코딩 방식은 UTF-8로 저저장
- HTML의 태그는 대부분 부등호(<>) 안에 씀
- HTML은 대문자와 소문자를 구분하지 않음
- HTML 파일은 <HTML> 태그로 시작해서 </HTML> 태그로 종료

## Section 02 웹 크롤링 기본

### ■ HTML 구조

```
<html>
```

```
<head>
```

화면에 표시되지 않는 정보(타이틀, 인코딩 정보 등을 표현)

```
</head>
```

```
<body>
```

화면에 표시되는 본체(주로 태그를 표현)

```
</body>
```

```
</html>
```

## Section 02 웹 크롤링 기본

### ■ HTML 구조

- `<html>~</html>`

전체 코드를 감쌈

- `<head>~</head>`

타이틀, 인코딩 정보 등 화면에 표시되지 않는 정보를 포함

- `<body>~</body>`

주로 화면에 표시되는 본체 및 태그를 포함

필요하다면 태그에 속성을 표시할 수 있음

- `<title>~</title>`

주로 `<head>` 속에서 웹 브라우저의 타이틀 바에 표시되는 웹 페이지의 제목을 표시

# Section 02 웹 크롤링 기본

## ■ HTML 태그

안녕하세요?  
데이터 분석을 학습 중입니다.  
이건 딸줄  
이건 쿡계  
이건 이탤릭

**폰트 변경했어요.**

[한빛 홈페이지 연결](#)

아이디	이름
BBK	바비킴
LSG	이승기

이 부분이 div 부분입니다. HTML을 코딩하면 됩니다.

- CSS
- 엑셀
- 데이터베이스

DIV에 id를 설정했어요.

DIV에 class를 설정했어요.

DIV에 class를 또 설정했어요.

2 두 줄로 출력된다.

4 궁서체 10pt의 빨간색 글자가 출력된다.

6 클릭하면 새로운 페이지에서 한빛의 홈페이지가 열린다.

8 제목에 아이디와 이름이 있는 3행 2열의 테이블이 출력된다.

그림 9-4 간단한 HTML 파일의 예

## Section 02 웹 크롤링 기본

### ■ HTML 태그

```
<table border=1>
<tr>
  <th>아이디</th>
  <th>이름</th>
</tr>
<tr>
  <td>BBK</td>
  <td>바비킴</td>
</tr>
<tr>
  <td>LSG</td>
  <td>이승기</td>
</tr>
</table>
```

## Section 02 웹 크롤링 기본

### ■ HTML 태그

- `<table>~</table>`, `<tr>~</tr>`, `<th>~</th>`, `<td>~</td>`

표를 만드는 태그들

`<table>~</table>` 태그 안에 행은 `<tr>~</tr>`로 구성되고, 행 안에 열이 `<th>~</th>` 또는 `<td>~</td>`로 구성

`<th>`는 제목 열을 표현해서 볼드체로 보이며 `<td>`는 일반 열로 표현됨

- `<div>~</div>`

Division의 약자이며, 웹 페이지에서 레이아웃을 만들 때 사용

필요한 부분을 묶어주는 역할을 하고, `style`을 지정해서 해당 부분을 표현할 수 있음



## Section 02 웹 크롤링 기본

### ■ HTML 태그

#### ■ id 속성과 class 속성

id 속성은 HTML 문서 중에서 유일한 값으로 하나의 태그에만 지정할 수 있음

class 속성은 여러 개의 태그에 지정할 수 있음

```
<div id = "myid" >  
    div에 id를 설정했어요.  
</div>  
<div class = "myclass" >  
    div에 class를 설정했어요.  
</div>  
<div class = "myclass" >  
    div에 class를 또 설정했어요.  
</div>
```

## Section 02 웹 크롤링 기본

### ■ 라이브러리 활용

- 설치할 패키지 이름은 bs4
- beautifulsoup 버전 4의 의미

```
pip install bs4
```

## Section 02 웹 크롤링 기본

### ■ 라이브러리 활용

- 웹 브라우저에서 네이트에 접속한 후, 본문 빈 곳에서 마우스 오른쪽 버튼을 클릭하고 [페이지 원본 보기]를 선택

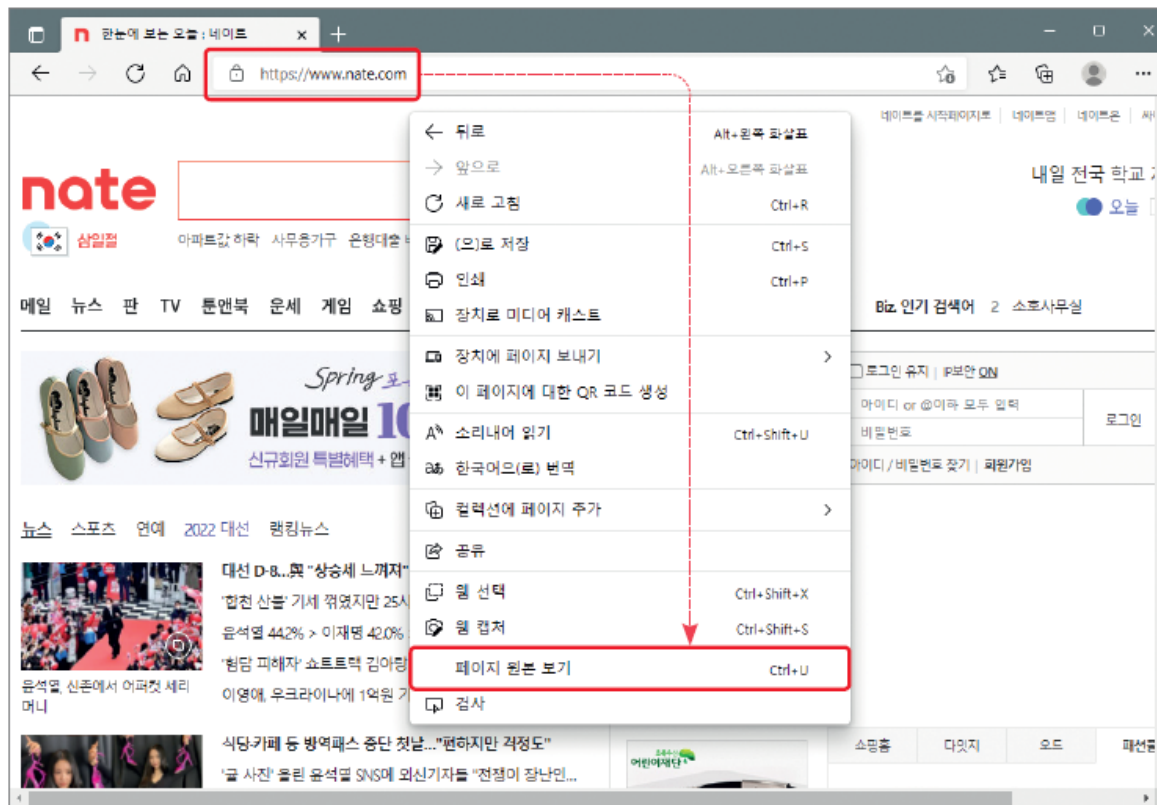
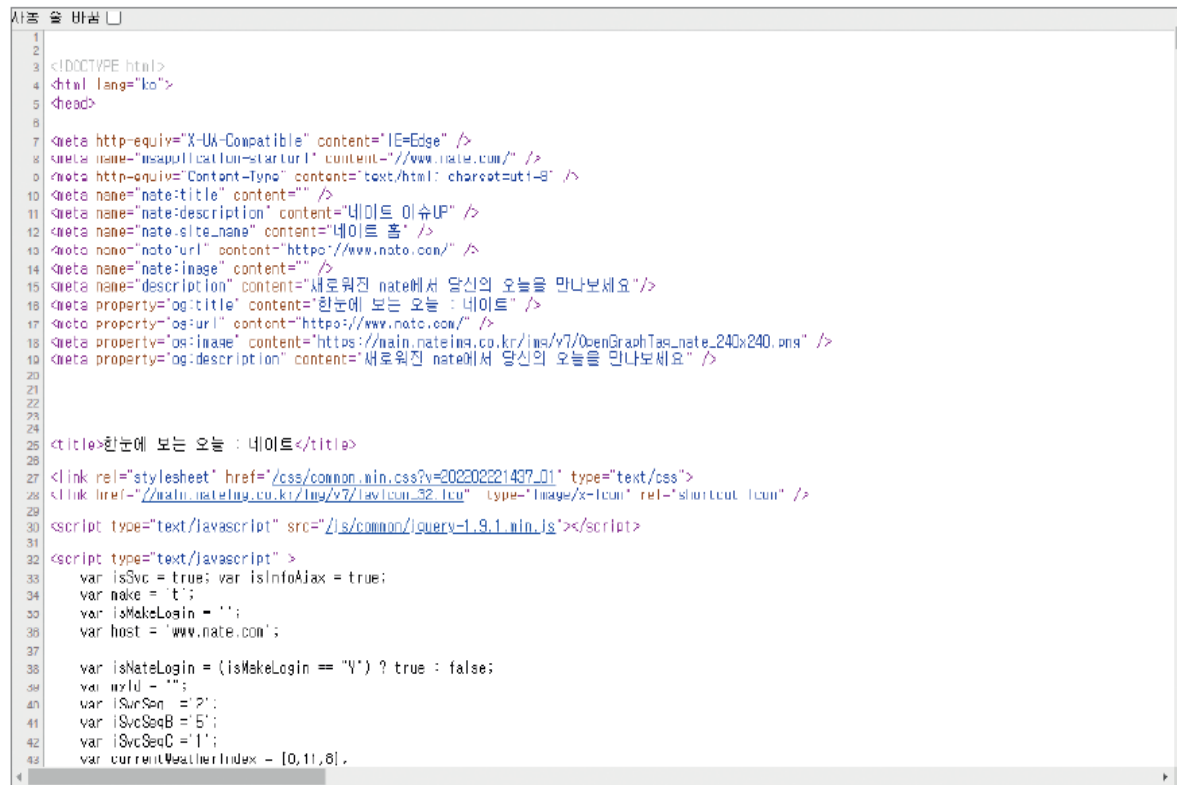


그림 9-5 HTML 소스코드 보기 1

## Section 02 웹 크롤링 기본

### ■ 라이브러리 활용

- 행 번호가 1000이 넘어갈 정도로 상당히 많은 분량의 HTML 코드로 작성되어 있는 것을 확인할 수 있음



```
1
2
3 <!DOCTYPE html>
4 <html lang="ko">
5 <head>
6
7 <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
8 <meta name="msapplication-starturl" content="//www.nate.com/" />
9 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
10 <meta name="nate:title" content="" />
11 <meta name="nate:description" content="네이트 이슈UP" />
12 <meta name="nate.site_name" content="네이트 홈" />
13 <meta name="nate:url" content="http://www.nate.com/" />
14 <meta name="nate:image" content="" />
15 <meta name="description" content="새로워진 nate에서 당신의 오늘을 만나보세요"/>
16 <meta property="og:title" content="한눈에 보는 오늘 : 네이트" />
17 <meta property="og:url" content="http://www.nate.com/" />
18 <meta property="og:image" content="https://main.nateimg.co.kr/img/v7/OcenGraphTas_nate_240x240.png" />
19 <meta property="og:description" content="새로워진 nate에서 당신의 오늘을 만나보세요" />
20
21
22
23
24
25 <title>한눈에 보는 오늘 : 네이트</title>
26
27 <link rel="stylesheet" href="/css/common.min.css?v=202202221437_01" type="text/css">
28 <link href="//main.nateimg.co.kr/img/v7/icon_32.ico" type="image/x-icon" rel="shortcut icon" />
29
30 <script type="text/javascript" src="/js/common/jquery-1.9.1.min.js"></script>
31
32 <script type="text/javascript">
33     var isSvc = true; var isInfoAjax = true;
34     var nake = 't';
35     var isMakeLogin = '';
36     var host = 'www.nate.com';
37
38     var isNateLogin = (isMakeLogin == "Y") ? true : false;
39     var nylid = '';
40     var isSvcSeq = '2';
41     var isSvcSeqB = '5';
42     var isSvcSeqC = '1';
43     var currentWeatherIndex = [0,11,8];
```

그림 9-6 HTML 소스코드 보기 2

## Section 02 웹 크롤링 기본

### ■ 라이브러리 활용

- 이러한 방식으로 HTML 코드에 접근하는 기능을 urllib.request가 제공함

Code09-01.py

```
01 import urllib.request
02
03 nateUrl = "https://www.nate.com"
04 htmlObject =
05 html = htmlObject.read()
06
07 print(html)
```

실행 결과

```
b'\r\n\r\n<!DOCTYPE html>\r\n<html lang="ko">\r\n<head>\r\n\t\r\n<meta http-equiv="X-UA-
Compatible" content="IE=Edge" /\r\n<meta name="msapplication-starturl" content="//www.
nate.com/" /\r\n<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>\r\n<meta name="nate:title" content="" /\r\n<meta name="nate:description" content="\
xeb\x84\xa4\xec\x9d\xb4\xed\x8a\xb8 \xec\x9d\xb4\xec\x8a
```

~~~ 생략 ~~~

## Section 02 웹 크롤링 기본

### ■ 라이브러리 활용

- 우리가 알아볼 수 있는 형태로 변경해주고 더불어 필요한 내용들을 추출하기 위해서는 BeautifulSoup을 사용해야 함

Code09-02.py

```
01 import urllib.request
02 import bs4
03
04 nateUrl = "https://www.nate.com"
05 htmlObject = urllib.request.urlopen(nateUrl)
06 bsObject = bs4.BeautifulSoup(htmlObject,
07
08 print(bsObject)
```

실행 결과

```
<!DOCTYPE html>

<html lang="ko">
<head>
<meta content="IE=Edge" http-equiv="X-UA-Compatible"/>
<meta content="//www.nate.com/" name="msapplication-starturl"/>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<meta content="" name="nate:title"/>
<meta content="네이트 이슈UP" name="nate:description"/>
<meta content="네이트 홈" name="nate:site_name"/>
~~~ 이하 생략 ~~~
```

## Section 02 웹 크롤링 기본

### ■ BeautifulSoup 사용 방법

- 앞서 출력한 HTML 코드는 너무 길고 비정형적 형태
- 긴 HTML 코드에서 필요한 부분을 추출하여 사용할 필요가 있음

### ■ HTML 코드 접근

Sample02.html

```
<html>
  <head>
  </head>
  <body>
    <div> 요기를 클릭하세요 </div>
    <ul>
      <li> 한빛출판네트워크 </li>
      <li> 비기너 </li>
      <li> 데이터 분석 </li>
    </ul>
  </body>
</html>
```

## Section 02 웹 크롤링 기본

### ■ HTML 코드 접근

- HTML 파일을 읽어서 BeautifulSoup를 통해 출력하는 코드

Code09-03.py

```
01 import bs4
02
03 webPage = open('C:/CookAnalysis/HTML/Sample02.html', 'rt',
04 encoding='utf-8').read()
05 bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
06
    print(bsObject)
```

실행 결과

```
<html>
<head>
</head>
<body>
<div> 요기를 클릭하세요 </div>
<ul>
<li> 한빛출판네트워크 </li>
<li> 비기너 </li>
<li> 데이터 분석 </li>
</ul>
</body>
```



## Section 02 웹 크롤링 기본

### ■ HTML 코드 접근

- BeautifulSoup의 목적인 필요한 내용을 추출한 후 <div> 태그의 내용을 찾아서 간단히 추출하는 코드

Code09-04.py

```
01 import bs4
02
03 webPage = open('C:/CookAnalysis/HTML/Sample02.html', 'rt',
04 encoding='utf-8').read()
05 bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
06
07 tag_div = bsObject.find('div')
   print(tag_div)
```

실행 결과

<div> 요기를 클릭하세요 </div>

## Section 02 웹 크롤링 기본

### ■ 여러 건의 데이터 추출

- 여러 건의 데이터가 들어있는 <ul> 태그의 모든 내용을 추출하는 코드

code09-05.py

```
01 import bs4
02
03 webPage = open('C:/CookAnalysis/HTML/Sample02.html', 'rt',
04 encoding='utf-8').read()
05 bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
06
07 tag_ul= bsObject.find('ul')
08 print(tag_ul)
09 print()
10
11 tag_li= bsObject.find('li')
12 print(tag_li)
13 print()
14
15 tag_li_all= bsObject.findAll('li')
   print(tag_li_all)
```

#### 실행 결과

```
<ul>
<li> 한빛출판네트워크 </li>
<li> 비기너 </li>
<li> 데이터 분석 </li>
</ul>
```

```
<li> 한빛출판네트워크 </li>
```

```
[<li> 한빛출판네트워크 </li>, <li> 비기너 </li>, <li> 데이터 분석 </li>]
```

## Section 02 웹 크롤링 기본

### ■ 특정 태그 추출

- id나 class를 지정해서 특정한 태그만 추출하는 방식을 사용할 수 있음
- find() 또는 findAll()을 다음과 같은 형식으로 사용

```
bsObject.find('태그명' , {'속성명' : '속성값'})  
bsObject.findAll('태그명' , {'속성명' : '속성값'})
```

## Section 02 웹 크롤링 기본

### ■ BeautifulSoup 사용 방법

#### ■ 샘플 코드를 준비

Sample03.html

```
<html>
  <head>
  </head>
  <body>
    <div id='myId1'> 아기공룡 </div>
    <div class='myClass1'> 내 친구 </div>
    <ul class='myClass2'>
      <li> 한빛아카데미 </li>
      <li> 한빛미디어 </li>
    </ul>
    <a href="www.daum.net"> 다음 바로가기 </a>
    <div class='myClass1'> 둘리 </div>
    <ul>
      <li class='myClass3'> 비기너 </li>
      <li class='myClass3'> 시리즈 </li>
    </ul>
    <a href="www.nate.com"> 네이트 바로가기
  </a>
    <a href="www.naver.com"> 네이버 바로가기
  </a>
  </body>
</html>
```

## Section 02 웹 크롤링 기본

### ■ BeautifulSoup 사용 방법

- find() 함수로는 id든 class든 1개씩만 추출되고, findAll()은 모든 해당 클래스를 모두 추출해서 리스트로 반환

Code09-06.py

```
01 import bs4
02
03 webPage = open('C:/CookAnalysis/HTML/Sample03.html', 'rt',
04 encoding='utf-8').read()
05 bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
06
07 tag = bsObject.find('div', {'id':'myId1'})
08 print(tag)
09
10 tag = bsObject.find('div', {'class':'myClass1'})
11 print(tag)
12
13 tag = bsObject.findAll('div', {'class':'myClass1'})
    print(tag)
```

실행 결과

```
<div id="myId1"> 아기공룡 </div>
<div class="myClass1"> 내 친구 </div>
[<div class="myClass1"> 내 친구 </div>, <div class="myClass1"> 둘리 </div>]
```

## Section 02 웹 크롤링 기본

### ■ BeautifulSoup 사용 방법

- class 값으로 <ul> 및 <li>의 값을 추출하고 출력

Code09-07.py

```
01 import bs4
02
03 webPage = open('C:/CookAnalysis/HTML/Sample03.html', 'rt',
04 encoding='utf-8').read()
05 bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
06
07 ul_value = bsObject.find('ul', {'class':'myClass2'})
08 print(ul_value)
09 print()
10 li_list = bsObject.findAll('li', {'class':'myClass3'})
   print(li_list)
```

#### 실행 결과

```
<ul class="myClass2">
<li> 한빛아카데미 </li>
<li> 한빛미디어 </li>
</ul>
```

```
[<li class="myClass3"> 비기너 </li>, <li class="myClass3"> 시리즈 </li>]
```

## Section 02 웹 크롤링 기본

### ■ BeautifulSoup 사용 방법

- <a>를 모두 추출한 후 <a>의 href 속성값인 URL을 모두 출력

code09-08.py

```
01 import bs4
02
03 webPage = open('C:/CookAnalysis/HTML/Sample03.html', 'rt',
04 encoding='utf-8').read()
05 bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
06
07 a_list = bsObject.findAll('a')
08 for aTag in a_list :
    print( aTag['href'] )
```

실행 결과

www.daum.net  
www.nate.com  
www.naver.com

## Section 02 웹 크롤링 기본

### ■ [프로그램 1] 완성

Code09-09.py

```
0  import bs4
1
0  webPage = open('C:/CookAnalysis/HTML/Sample02.html', 'rt',
2  encoding='utf-8').read()
0  bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
3
0  tag_li_all= bsObject.findAll('li')
4  for tag_li in tag_li_all :
0      print(tag_li.text)
5  print()
0  for i in range(len(tag_li_all)) :
6      print(tag_li_all[i].text)
7
0
8
0
9
```

#### 실행 결과

한빛출판네트워크  
비기너  
데이터 분석

한빛출판네트워크  
비기너  
데이터 분석



## Section 03 웹 크롤링 활용

### ■ 웹 사이트 정보 추출

- 웹 페이지 기본 정보 알아내기
  - 엣지 브라우저를 실행해서 네이트에 접속
  - 오른쪽 위 [...] 모양의 '설정 및 기타' 아이콘을 클릭해서 확장한 후 [기타]-[개발자 도구]를 선택

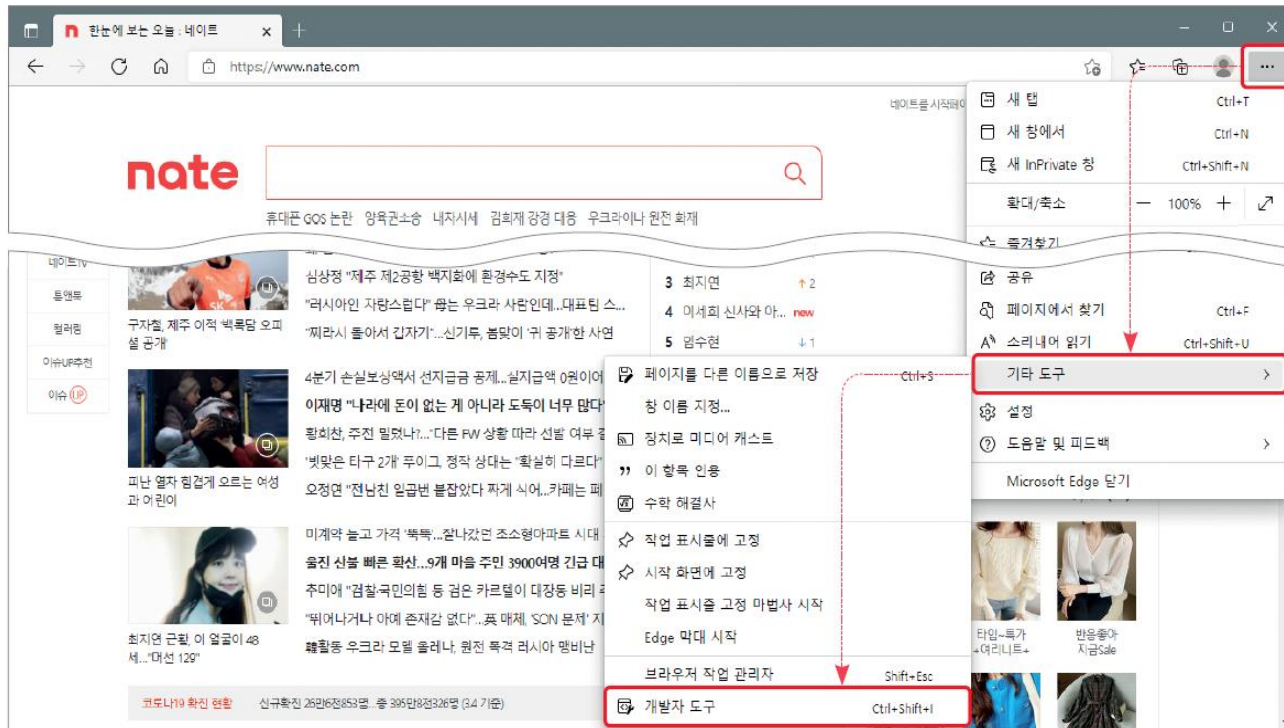


그림 9-7 개발자 도구 사용 1

## Section 03 웹 크롤링 활용

### ■ 웹 사이트 정보 추출

- 웹 페이지 기본 정보 알아내기
  - [검사할 페이지 요소를 선택하세요] 아이콘을 클릭하고 왼쪽 화면에서 찾고자 하는 부분을 클릭
  - 오른쪽에 해당 부분의 소스가 표시됨
  - 왼쪽 화면에는 풍선 도움말로 선택한 부분의 정보가 나옴

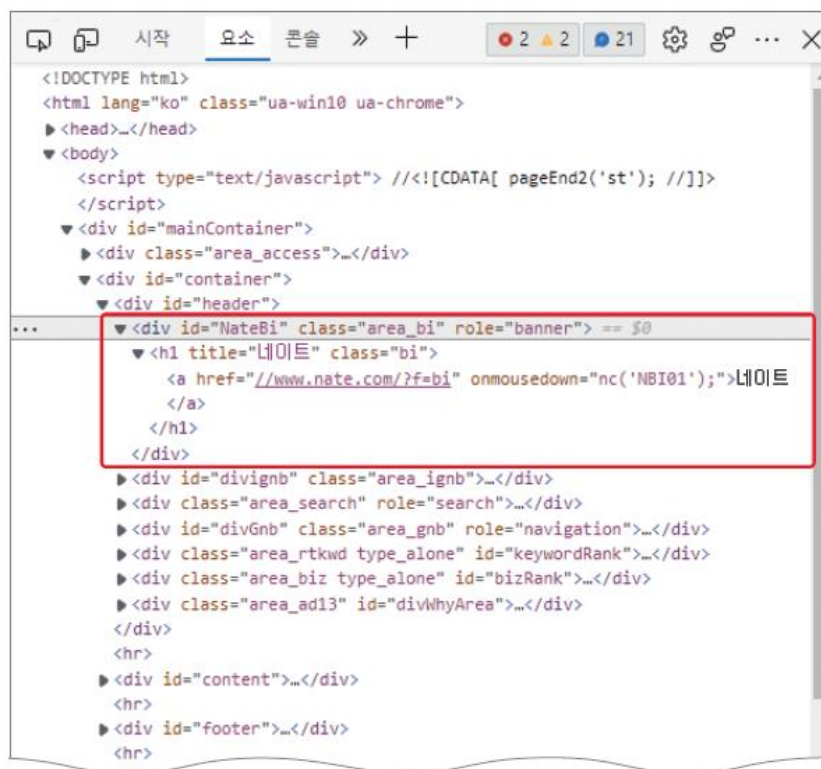


그림 9-8 개발자 도구 사용 2

## Section 03 웹 크롤링 활용

### ■ 웹 사이트 정보 추출

- 웹 페이지 기본 정보 알아내기
  - 오른쪽 <div> 부분을 확장하면 해당 부분의 소스 위치를 찾을 수 있음



## Section 03 웹 크롤링 활용

### ■ 웹 사이트 정보 추출

- 네이트 사이트의 로고를 클릭할 때 연결되는 주소 및 로고에 지정된 글자를 추출하는 코드

Code09-10.py

```
01 import bs4
02 import urllib.request
03
04 nateUrl = "https://www.nate.com"
05 htmlObject = urllib.request.urlopen(nateUrl)
06 webPage = htmlObject.read()
07 bsObject = bs4.BeautifulSoup(webPage, 'html.parser')
08
09 tag = bsObject.find('div', {'id': 'NateBi'})
10 print(tag, '\n')
11
12 a_tag = tag.find("a")
13 print(a_tag, '\n')
14
15 href = a_tag['href']
16 print(href, '\n')
17
18 text = a_tag.text
19 print(text)
```

실행 결과

```
<div class="area_bi" id="NateBi" role="banner">
<h1 class="bi" title="네이트"><a href="//www.nate.com/?f=bi" onmousedown="nc('NBI01');">네이
트</a></h1>
</div>

<a href="//www.nate.com/?f=bi" onmousedown="nc('NBI01');">네이트</a>

//www.nate.com/?f=bi

네이트
```

## Section 03 웹 크롤링 활용

### ■ 웹 사이트 정보 추출

#### ■ 웹 페이지 메뉴 알아내기

- 네이트 뉴스(<https://news.nate.com>)의 메뉴를 추출
- 네이트 뉴스에 접속한 후 개발자 도구로 다음 그림과 같이 메뉴 부분을 클릭해서 코드를 확인

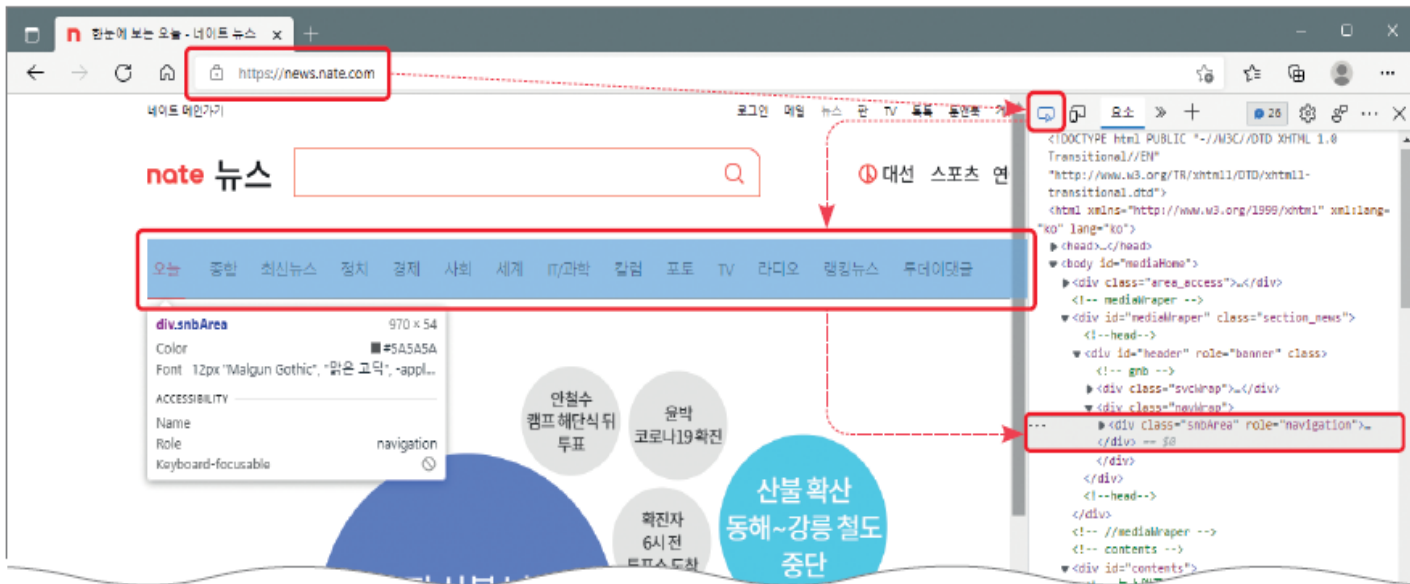


그림 9-10 네이트 뉴스의 메뉴 추출 1

## Section 03 웹 크롤링 활용

### ■ 웹 사이트 정보 추출

#### ■ 웹 페이지 메뉴 알아내기

- 오른쪽 HTML 코드에서 <div class="snbArea" ~> 를 찾을 수 있음
- 이 부분을 확장하면 해당 메뉴의 목록을 확인할 수 있음

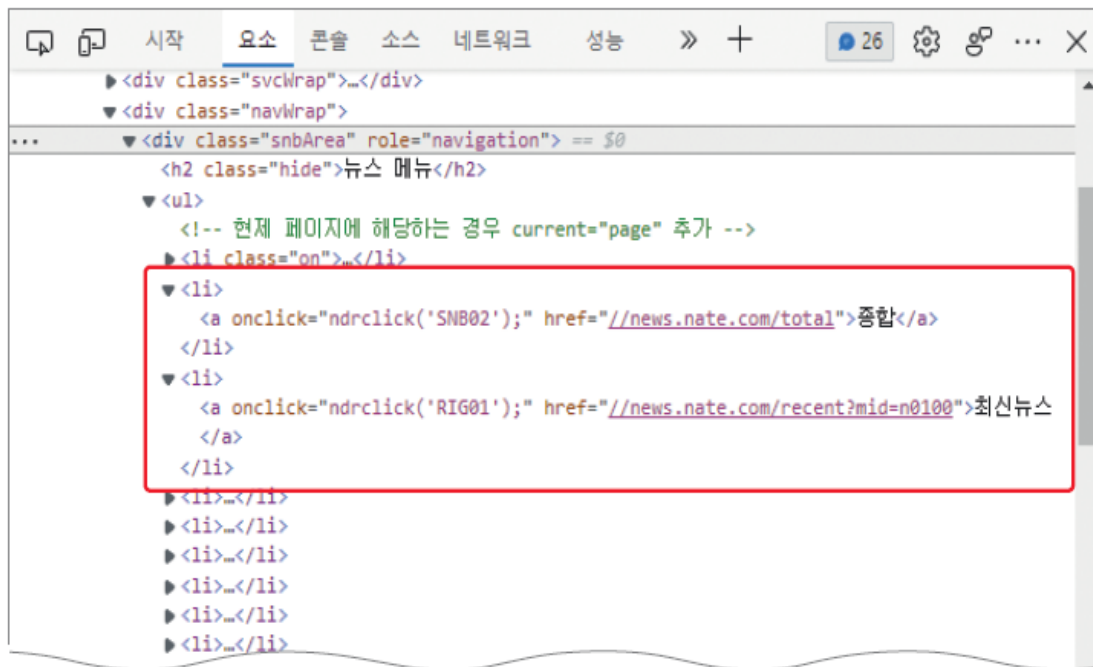


그림 9-11 네이트 뉴스의 메뉴 추출 2

## Section 03 웹 크롤링 활용

### ■ 웹 사이트 정보 추출

#### ■ 웹 페이지 메뉴 알아내기

- <div> 태그의 “snbArea” 클래스를 추출한 후에, 그 안의 <li> 태그의 텍스트를 추출하면 뉴스 메뉴의 목록이 됨

#### Code09-11.py

```
01  ~~ Code09-10.py의 1~7행과 동일. 단 주소는
...  "https://news.nate.com" ~~
08
09  tag = bsObject.find('div', {'class':'snbArea'})
10
11  print('## 네이트 뉴스의 메뉴 목록 ##')
12  li_list = tag.findAll('li')
13  for <li> in li_list :
        print(li.text, end=' ' )
```

#### 실행 결과

## 네이트 뉴스의 메뉴 목록 ##

오늘 종합 최신뉴스 정치 경제 사회 세계 IT/과학 칼럼 포토 TV 라디오 랭킹뉴스 투데이댓글