

데이터 크롤링과 정제

1장. 첫 번째 웹 스크레이퍼

목차

- 웹 스크레이핑과 웹 크롤링 개념
- BeautifulSoup 소개
- HTML 구성 및 태그
- CSS 구성

개념 정리

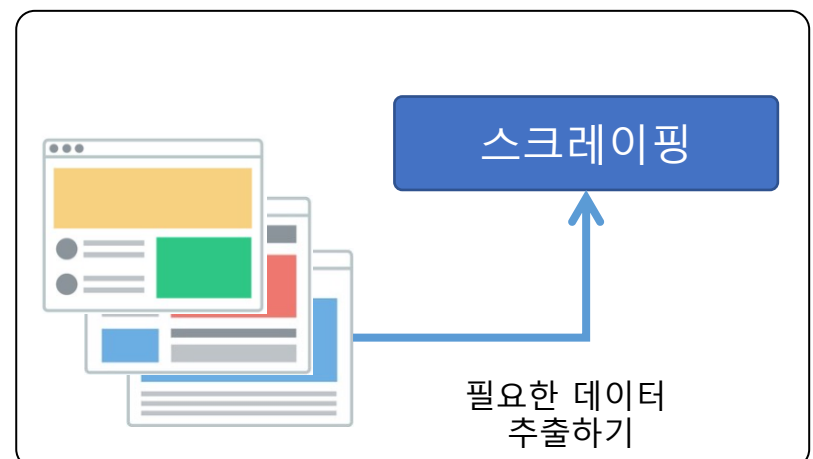
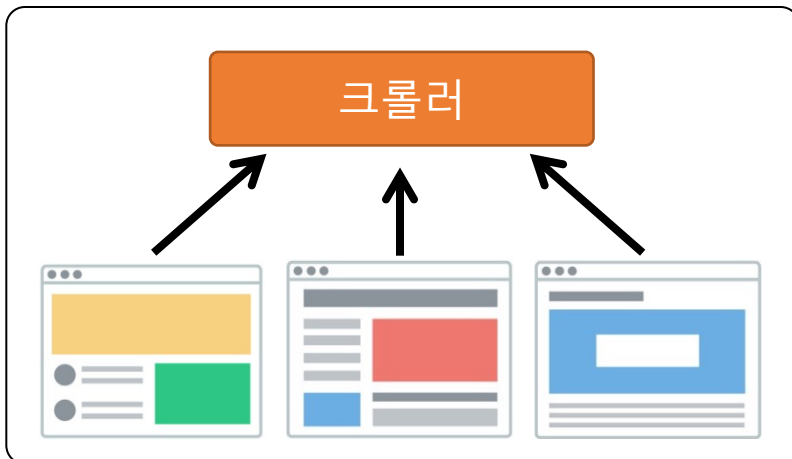
■ 크롤링(Crawling)과 스크레이핑(Scraping)

• 크롤러

- 자동으로 웹 페이지에 있는 정보를 수집하는 프로그램
- 크롤링: 웹 크롤러로 정보를 수집하는 일
- Google 등의 검색 엔진에서 정보를 검색하는 방식
 - 전 세계에 있는 웹 페이지 정보를 모아서 축적

• 스크레이핑

- 수집한 정보를 분석해서 필요한 정보를 추출
- 전자 상거래 사이트에서 웹 크롤러로 다운 받음
 - 웹페이지에서 상품 이름과 가격 등의 필요한 정보를 추출



웹 페이지 가져오기

- `urllib.request.urlopen(url)`
 - 해당 url에서 HTML파일이나 이미지 파일, 기타 파일을 가져오는 함수
 - 리턴값: `HTTPResponse` 객체
- `HTTPResponse.read()`
 - HTML 콘텐츠를 읽어옴 (리턴값: bytes 형태)

```
from urllib.request import urlopen

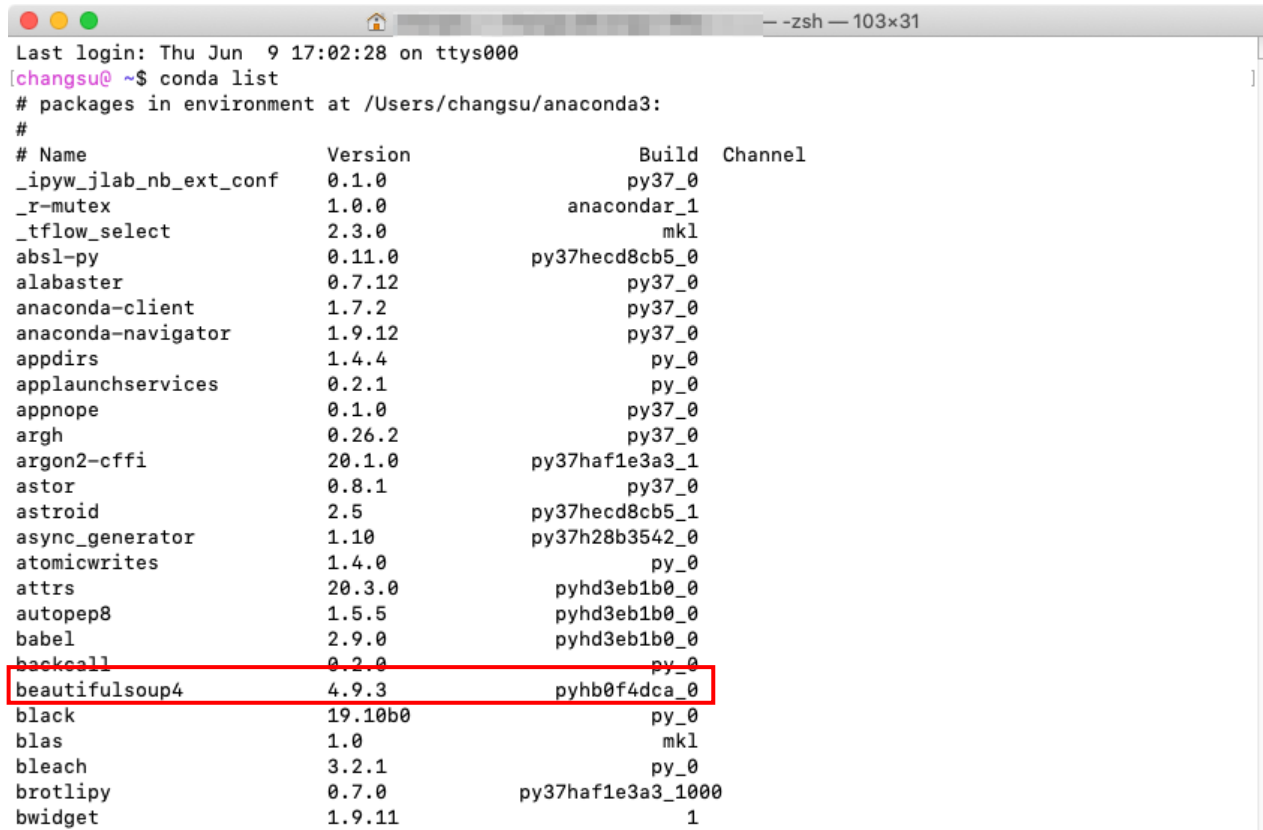
html = urlopen('https://www.daangn.com/hot_articles')
print(type(html))
print(html.read())
```

```
<class 'http.client.HTTPResponse'>
b'<!DOCTYPE html>\n<html lang="ko">\n<head>\n  <meta charset="utf-8">\n  <meta http-
equiv="X-UA-Compatible" content="IE=edge">\n  <meta name="viewport"
content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no">\n  \n
<link rel="canonical" href="https://www.daangn.com/hot_articles" />\n\n
<title>\xeb\x8b\xb9\xea\xb7\xbc\xeb\xa7\x88\xec\xbc\x93
\xec\xa4\x91\xea\xb3\xa0\xea\xb1\xb0\xeb\x9e\x98 | \xeb\x8b\xb9\xec\x8b\xa0
\xea\xb7\xbc\xec\xb2\x98\xec\x9d\x98
. . .
```

BeautifulSoup4 라이브러리 설치 확인

■ BeautifulSoup 라이브러리 확인

- 잘못된 HTML을 수정하여 쉽게 탐색할 수 있는 XML 형식의 파이썬 객체로 변환
- 아나콘다 패키지에 포함
- `$ conda list` 명령어로 설치 확인



```
Last login: Thu Jun  9 17:02:28 on ttys000
[changsu@ ~]$ conda list
# packages in environment at /Users/changsu/anaconda3:
#
# Name                    Version            Build    Channel
_ipyw_jlab_nb_ext_conf    0.1.0              py37_0
_r-mutex                  1.0.0              anacondar_1
_tflow_select             2.3.0              mkl
absl-py                   0.11.0             py37hecd8cb5_0
alabaster                  0.7.12             py37_0
anaconda-client            1.7.2              py37_0
anaconda-navigator         1.9.12             py37_0
appdirs                    1.4.4              py_0
applaunchservices          0.2.1              py_0
appnope                    0.1.0              py37_0
argh                       0.26.2             py37_0
argon2-cffi                20.1.0             py37haf1e3a3_1
astor                      0.8.1              py37_0
astroid                    2.5                py37hecd8cb5_1
async_generator            1.10               py37h28b3542_0
atomicwrites               1.4.0              py_0
attrs                      20.3.0             pyhd3eb1b0_0
autopep8                   1.5.5              pyhd3eb1b0_0
babel                      2.9.0              pyhd3eb1b0_0
backcall                   0.2.0              py_0
beautifulsoup4             4.9.3              pyhb0f4dca_0
black                      19.10b0            py_0
blas                       1.0                mkl
bleach                     3.2.1              py_0
brotlipy                   0.7.0              py37haf1e3a3_1000
bwidget                    1.9.11             1
```

BeautifulSoup 라이브러리

■ BeautifulSoup 객체 구조

```
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('http://www.pythonscraping.com/pages/page1.html')
bs = BeautifulSoup(html.read(), 'html.parser')
print(bs)
```

html

head
title

body

h1

div

```
<html>
<head>
<title>A Useful Page</title>
</head>
<body>
<h1>An Interesting Title</h1>
<div>
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</div>
</body>
</html>
```

실패할 수 있는 연결과 예외 처리

■ 예외 처리

- 페이지를 찾을 수 없는 경우
 - 404 Page Not Found 에러 발생: `HTTPError` 예외 발생 시킴
- 서버를 찾을 수 없는 경우
 - 500 Internal Server Error 발생시 `URLError` 예외 발생 시킴

```
from urllib.request import urlopen
from urllib.error import HTTPError
from urllib.error import URLError

try:
    html = urlopen('http://www.pythonscraping.com/pages/error.html')
except HTTPError as e:
    print(e)
except URLError as e:
    print('The server could not be found!')
else:
    print('It worked!')
```

HTTP Error 404: Not Found

존재하지 않는 태그 예외 처리

- 존재하지 않는 태그 접근
 - None 객체 반환
 - None 객체에 접근: `AttributeError` 발생

```
from urllib.request import urlopen
from urllib.error import HTTPError
from bs4 import BeautifulSoup

def getTitle(url, tag):
    try:
        html = urlopen(url)
    except HTTPError as e:
        return None
    try:
        bsObj = BeautifulSoup(html.read(), 'html.parser')
        value = bsObj.body.find(tag)
    except AttributeError as e:
        return None
    return value

tag = 'h2'
value = getTitle("http://www.pythonscraping.com/pages/page1.html", tag)
if value == None:
    print('{0} could not be found'.format(tag))
else:
    print(value)
```

h2 could not be found

HTML & CSS

HTML 구조

■ HTML 기본 구조

```
<!DOCTYPE html>
```

<!DOCTYPE html>
- 문서 형식 선언

```
<html>
```

```
<head>
```

```
<title>A Useful Page</title>
```

```
</head>
```

```
<body>
```

```
<h1>An Interesting Title</h1>
```

```
<div>
```

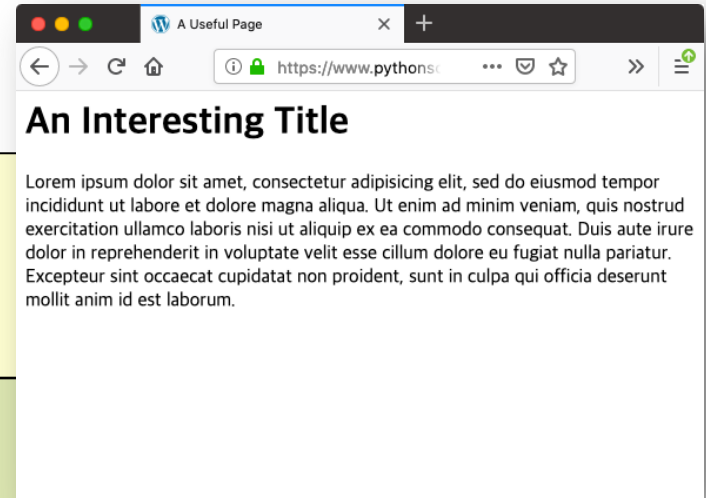
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
</div>
```

```
</body>
```

```
</html>
```

<body> ... </body>
- HTML 문서의 텍스트,
하이퍼링크, 이미지 등
콘텐츠를 포함하는 영역



HTML 글자 태그

■ 제목 나타내기

- <h1> ~<h6>

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>제목 연습(h1 ~ h6)</title>
</head>
<body>
<h1>글자 제목(h1)</h1>
<h2>글자 제목(h2)</h2>
<h3>글자 제목(h3)</h3>
<h4>글자 제목(h4)</h4>
<h5>글자 제목(h5)</h5>
<h6>글자 제목(h6)</h6>
</body>
</html>
```

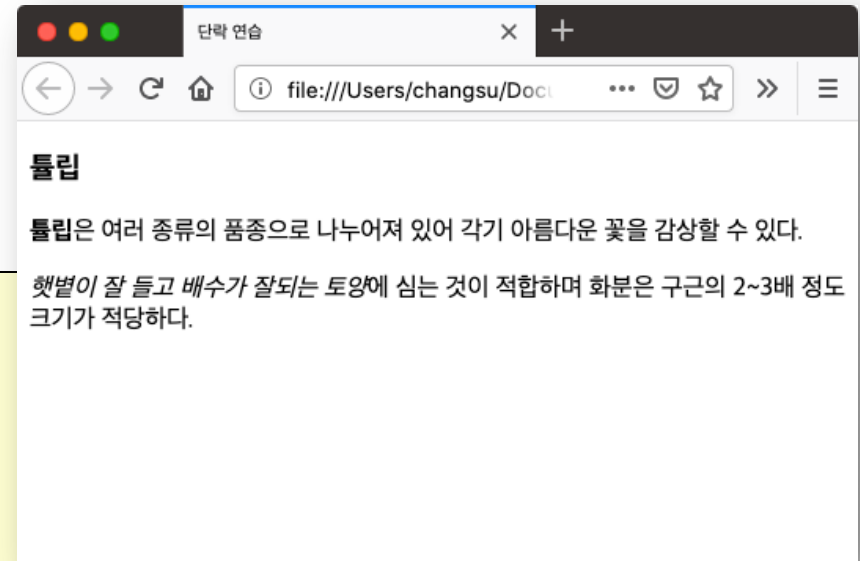
<meta> 태그
- 데이터 표현 속성
- 표준 문자 세트 UTF-8



단락 구분 태그: <p> ... </p>

- <p> ... </p> 태그
 - 단락 구분

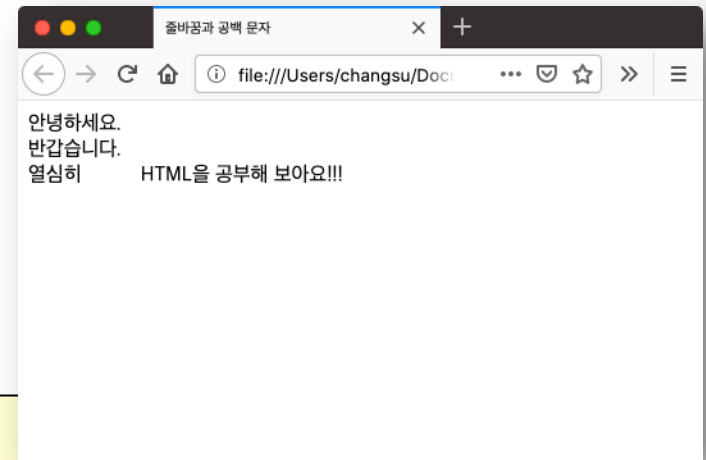
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>단락 연습</title>
</head>
<body>
<h3>튤립</h3>
<p><b>튤립</b>은 여러 종류의 품종으로 나누어져 있어 각기 아름다운 꽃을 감상할 수 있다.</p>
<p><i>햇별이 잘 들고 배수가 잘되는 토양</i>에 심는 것이 적합하며 화분은 구근의 2~3배 정도 크기가 적당하다.</p>
</body>
</html>
```



줄 바꿈과 공백

- `
` 태그
 - 줄 바꿈 태그
- ` `
 - 공백 문자 (non-breaking space의 약자)
 - 스페이스는 개수와 상관없이 1개만 표시

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>줄바꿈과 공백 문자</title>
</head>
<body>
안녕하세요.<br>
반갑습니다.<br>
열심히 &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; HTML을 공부해 보아요!!!
</body>
</html>
```

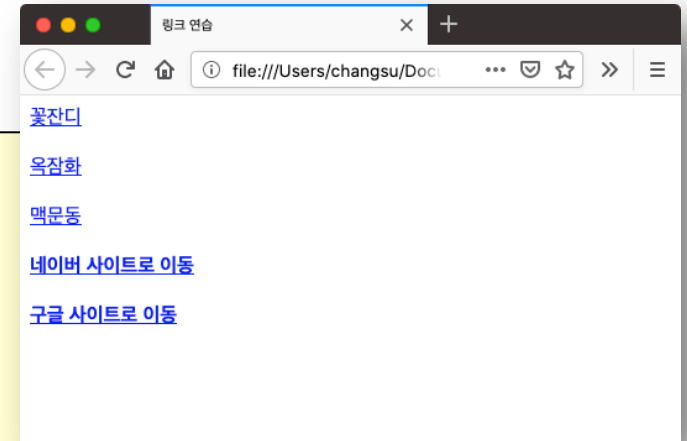


링크 태그

■ <a> 태그

- 웹 페이지에서 메뉴, 배너, 이미지 등을 클릭하면 지정된 페이지로 이동
- href 속성: 이동할 경로를 설정

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>링크 연습</title>
</head>
<body>
<a href="page1.html">꽃잔디</a><br><br>
<a href="page2.html">옥잠화</a><br><br>
<a href="page3.html">맥문동</a><br><br>
<a href="http://naver.com" target="_blank"><b>네이버 사이트로
이동</b></a><br><br>
<a href="http://google.com" target="_blank"><b>구글 사이트로 이동</b></a>
</body>
</html>
```



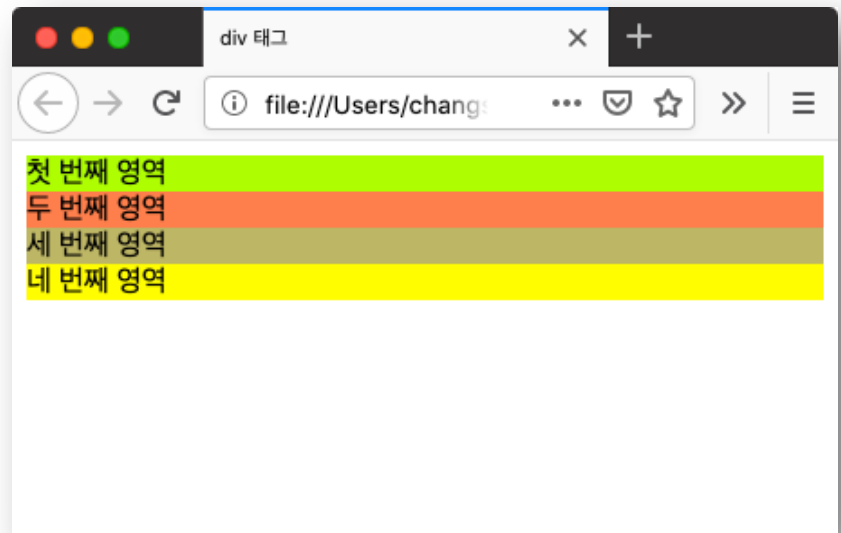
공간 분할 태그: div

■ <div> 태그

- 블록 형식으로 공간을 분할: division의 약자
 - 줄 바꿈이 가능함

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>div 태그 </title>
  <style>
    #section1 {
      background-color: greenyellow;
    }
    #section2 {
      background-color: coral;
    }
    #section3 {
      background-color: darkkhaki;
    }
    #section4 {
      background-color: yellow;
    }
  </style>
</head>
```

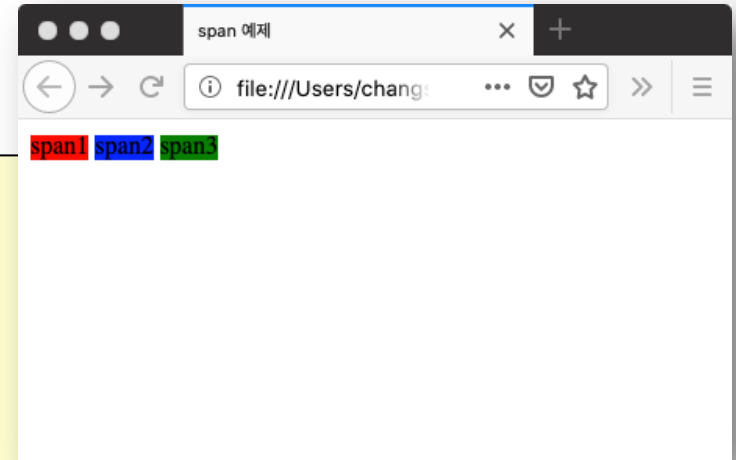
```
<body>
  <div id="section1">첫 번째 영역 </div>
  <div id="section2">두 번째 영역 </div>
  <div id="section3">세 번째 영역 </div>
  <div id="section4">네 번째 영역 </div>
</body>
</html>
```



span 태그

- 태그
 - 인라인(inline) 형식으로 공간 분할
 - 줄 바꿈이 되지 않음

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>span 예제 </title>
</head>
<body>
<html>
  <span style="background-color:red">span1</span>
  <span style="background-color:blue">span2</span>
  <span style="background-color:green">span3</span>
</html>
</body>
</html>
```



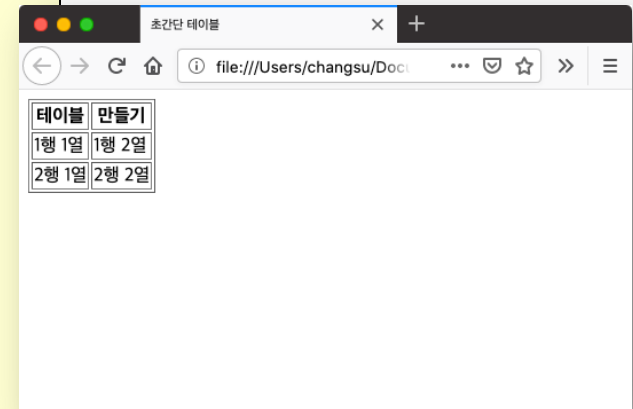
HTML Table 구성 태그

HTML 테이블 구성 요소	Tag	내용
	<code><table></table></code>	테이블을 만드는 태그
	<code><th></th></code>	테이블 헤더 부분 태그 (table header)
	<code><tr></tr></code>	테이블 행을 만드는 태그 (table row)
	<code><td></td></code>	테이블 열을 만드는 태그 (table data)

```

<html>
<head> <meta charset="UTF-8">
<title>초간단 테이블</title>
</head>
<body>
  <table border="1">
    <tr>
      <th>테이블</th>
      <th>만들기</th>
    </tr>
    <tr>
      <td>1행 1열</td>
      <td>1행 2열</td>
    </tr>
    <tr>
      <td>2행 1열</td>
      <td>2행 2열</td>
    </tr>
  </table>
</body>
</html>

```



```

<tr>
  <th>테이블</th>
  <th>만들기</th>
</tr>
<tr>
  <td>1행 1열</td>
  <td>1행 2열</td>
</tr>
<tr>
  <td>2행 1열</td>
  <td>2행 2열</td>
</tr>

```

HTML 태그 구성

- HTML 태그 및 속성
 - 태그 구성
 - 태그(tag)
 - 속성(attribute)
 - 속성값(value)



고급 HTML 분석: CSS 스타일

■ HTML 분석

- CSS 속성을 이용한 태그 검색

- CSS(Cascading Style Sheets) 요소 활용

- CSS 개요

- 웹 페이지 스타일 및 레이아웃에 사용

- 콘텐츠의 글꼴, 색상, 크기 및 간격을 변경 등

- class와 id 속성 사용

```
#아이디{ 속성1:속성값; 속성2:속성값; }
```

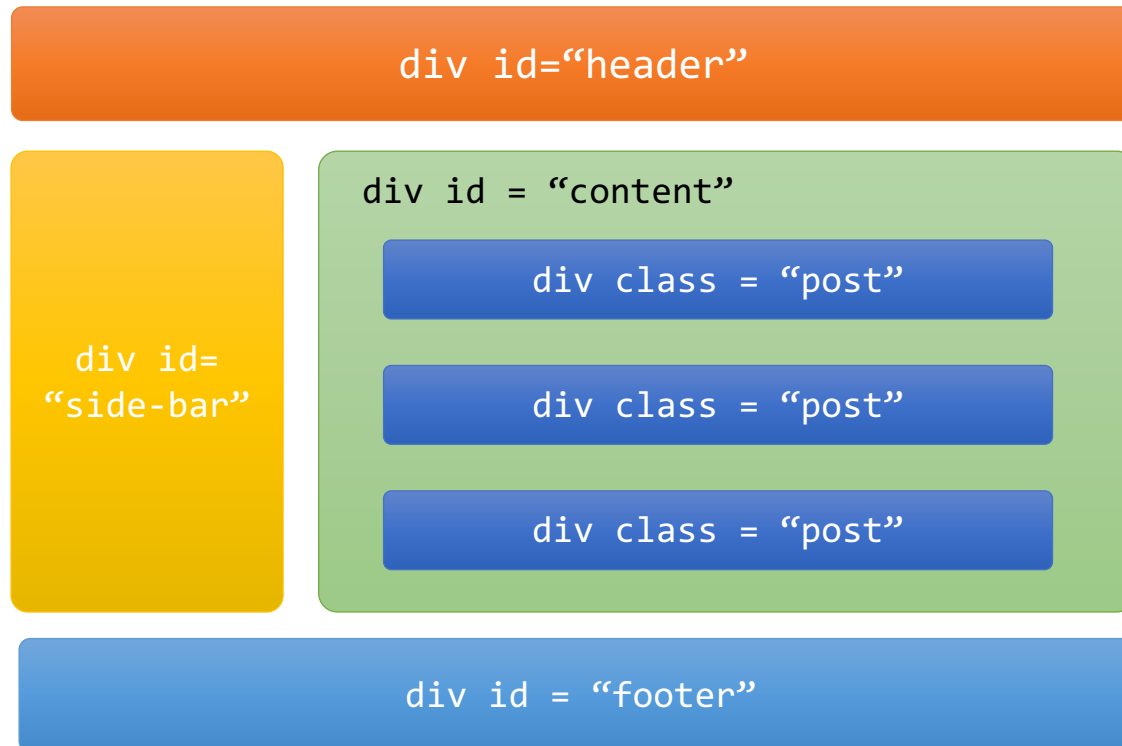
```
.클래스명{ 속성1:속성값; 속성2:속성값 }
```

```
<html>
<head>
<style>
    #m_box{ background-color: #09C; width: 150px; height: 40px; }
    .box{ width: 100px; height: 50px; border: 1px solid green }
</style>
</head>
<body>
    <div class="box">box 클래스</div>
    <div class="box">box 클래스</div>
    <div id="m_box">m_box 아이디</div>
</body>
</html>
```

HTML CSS class, id 차이

■ class와 Id 차이점

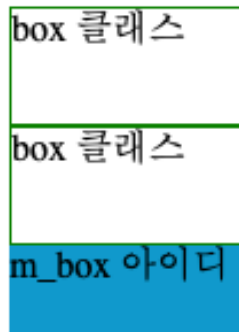
- id: 한 페이지에 한 요소에만 사용
 - 접근 방법: `#id명`
- class: 여러 요소에 중복 사용 가능한 스타일 지정
 - 반복적으로 사용되는 스타일에 class를 이용하여 정의
 - 접근 방법: `.class명`




CSS

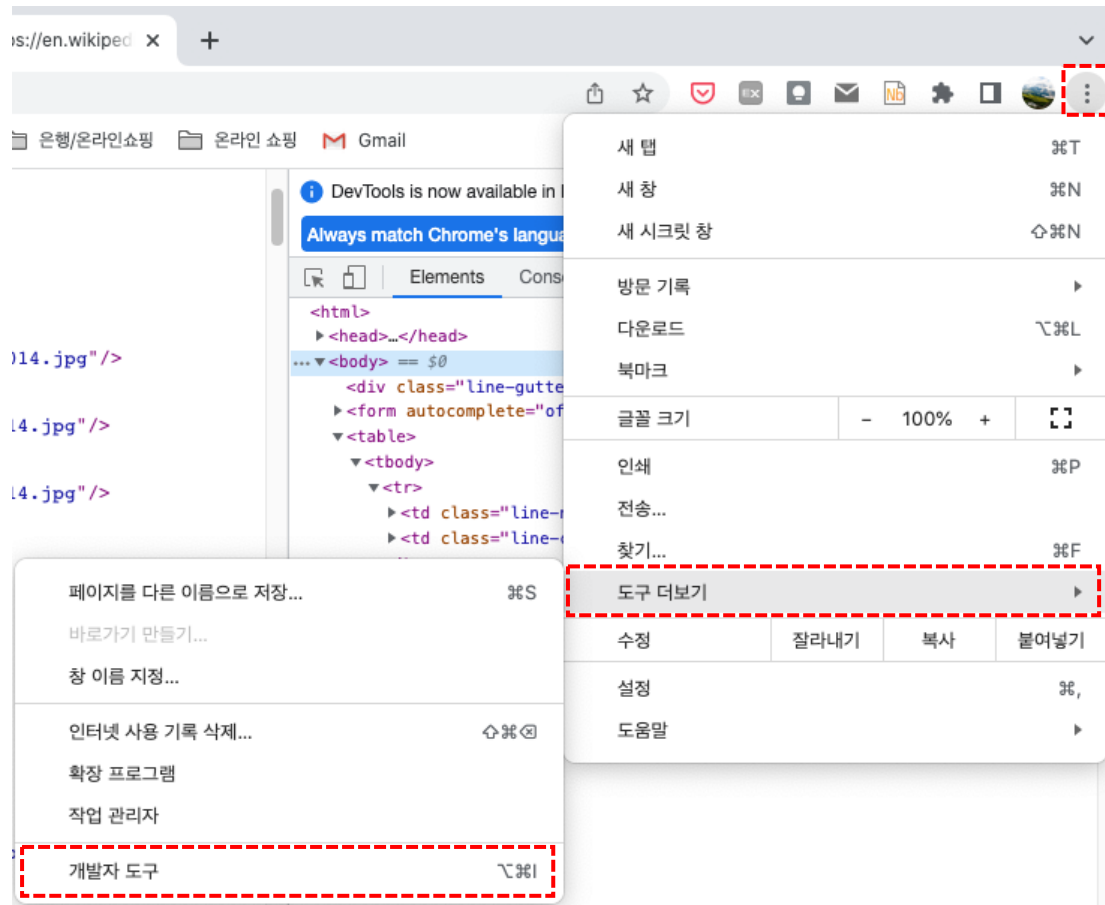
■ css class 간단 예제

```
<html>
<head>
<style>
  m_box{ background-color: #09C; width: 150px; height: 40px; }
  .box{ width: 100px; height: 50px; border: 1px solid green }
</style>
</head>
<body>
  <div class="box">box 클래스</div>
  <div class="box">box 클래스</div>
  <div id="m_box">m_box 아이디</div>
</body>
</html>
```

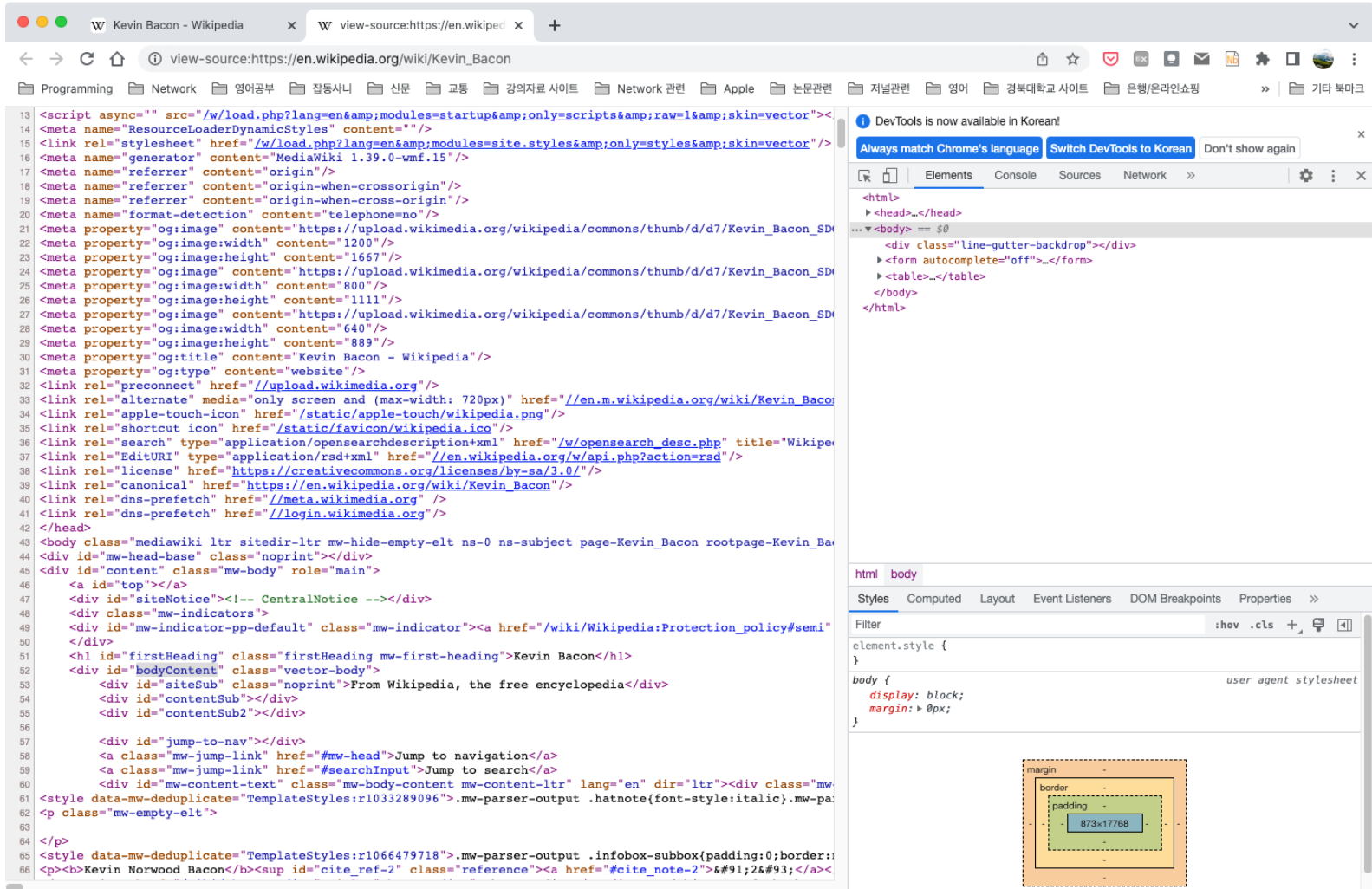


Chrome 웹브라우저에서 HTML 보기

- 오른쪽 상단의  메뉴 선택
- 도구 더보기 > 개발자 도구 선택



Chrome 웹브라우저에서 HTML 보기



참고 자료

마크업 언어와 마크다운 언어

■ 마크업(Markup) 언어

- Mark(태그)로 둘러싸인 언어
- 태그는 문서의 골격을 작성하는데 사용
- HTML, XML 등

```
1 <html>
2 <head>
3 <title>A Useful Page</title>
4 </head>
5 <body>
6 <h1>An Interesting Title</h1>
7 <div>
8 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore
9 ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate v
10 occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
11 </div>
12 </body>
13 </html>
```

■ 마크다운(Markdown)언어

- 마트업 언어의 일종
- 읽기 및 쓰기가 쉬운 문서 양식
 - 복잡한 태그 구조가 사라지고 간단한 텍스트들과 몇 가지 문법으로 작성

#: 페이지 헤딩
*: 순서가 없는 리스트 작성
[글씨]: 기울인 글씨
[링크내용](링크주소): 링크 생성

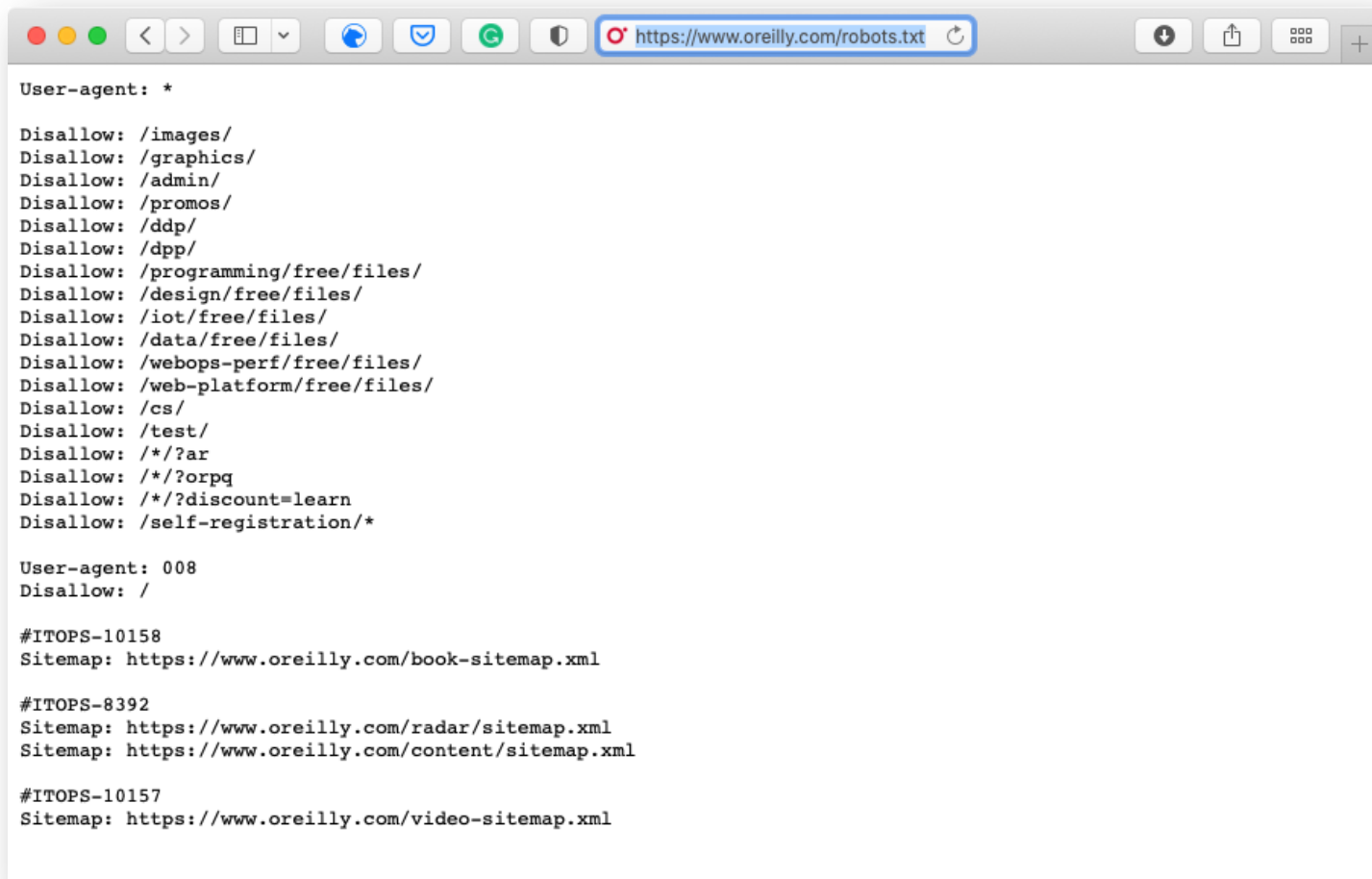
urllib.request와 requests 차이점

■ 차이점

	requests	urllib.request
차이점	<ul style="list-style-type: none">· 데이터를 딕셔너리 형태로 전송· 없는 페이지 요청시 에러 발생하지 않음· 기본 패키지가 아님 (추가 설치)	<ul style="list-style-type: none">· 데이터를 바이너리 형태로 전송· 없는 페이지 요청시 에러 발생· 기본 파이썬 패키지
사용법	<pre>import requests from bs4 import BeautifulSoup html = requests.get(url) soup = BeautifulSoup(html.text, 'html.parser')</pre>	<pre>import urllib.request import urlopen from bs4 import BeautifulSoup html = urlopen(url) soup = BeautifulSoup(html.read(), 'html.parser')</pre>
속성	<ul style="list-style-type: none">· .content 속성: 디코딩하지 않은 바이너리 형식의 데이터· .text 속성: utf-8로 인코딩된 문자열	<ul style="list-style-type: none">· urlopen(): HTTPResponse 객체 리턴· read(): 바이트 형태의 html을 읽어옴

웹 크롤링 가능 여부 확인

- robots.txt 추가
 - 웹사이트의 URL + **/robots.txt** 추가
 - 예: <https://www.oreilly.com/robots.txt>

A screenshot of a web browser window displaying the robots.txt file for the website https://www.oreilly.com. The browser's address bar shows the URL. The page content is a text-based file with various directives. The first section is for 'User-agent: *' and lists numerous paths that are disallowed, including /images/, /graphics/, /admin/, /promos/, /ddp/, /dpp/, and several /free/files/ directories. The second section is for 'User-agent: 008' and disallows the root path '/'. The third section, '#ITOPS-10158', lists sitemaps for book, radar, and content. The fourth section, '#ITOPS-8392', lists sitemaps for radar and content. The fifth section, '#ITOPS-10157', lists a video sitemap.

```
User-agent: *  
  
Disallow: /images/  
Disallow: /graphics/  
Disallow: /admin/  
Disallow: /promos/  
Disallow: /ddp/  
Disallow: /dpp/  
Disallow: /programming/free/files/  
Disallow: /design/free/files/  
Disallow: /iot/free/files/  
Disallow: /data/free/files/  
Disallow: /webops-perf/free/files/  
Disallow: /web-platform/free/files/  
Disallow: /cs/  
Disallow: /test/  
Disallow: /*/?ar  
Disallow: /*/?orpq  
Disallow: /*/?discount=learn  
Disallow: /self-registration/*  
  
User-agent: 008  
Disallow: /  
  
#ITOPS-10158  
Sitemap: https://www.oreilly.com/book-sitemap.xml  
  
#ITOPS-8392  
Sitemap: https://www.oreilly.com/radar/sitemap.xml  
Sitemap: https://www.oreilly.com/content/sitemap.xml  
  
#ITOPS-10157  
Sitemap: https://www.oreilly.com/video-sitemap.xml
```



Questions?