2021년도
ICT 이노베이션스퀘어 확산 사업

[인공지능 : 기초부터 실전까지]

▸ 14차수 ~ 16차수 강의 교안

※ 본 교안은 강의 수강 용도로만 사용 가능합니다.
상업적 이용을 일절 금함.

사람투자 과학기술정보통신부 nipa 정보통신산업진흥원 National IT Industry Promotion Agency KSA 한국표준협회

14

웹 스크레이핑

엑셀 파일 읽고 쓰기

- 웹 사이트에 접속해 필요한 자료를 수집하고 정리하는 작업
- 컴퓨터 소프트웨어 기술을 활용해 웹 사이트 내에 있는 정보를 추출하는 일

웹 브라우저로 웹 사이트 접속하기

- 내장 모듈인 webbrowser를 통해 웹 브라우저를 열고 지정된 웹 사이트에 접속
- 하나의 웹 사이트에 접속하기

```
In: import webbrowser

    url = 'www.naver.com'
    webbrowser.open(url)
Out: True
```

```
In: import webbrowser

    naver_search_url = "http://search.naver.com/search.naver?query="
    search_word = '파이썬'
    url = naver_search_url + search_word

    webbrowser.open_new(url)
Out: True
```

웹 브라우저로 웹 사이트 접속하기

- 하나의 웹 사이트에 접속하기

```
In: import webbrowser
    google_url = "www.google.com/#q="
    search_word = 'python'
    url = google_url + search_word
    webbrowser.open_new(url)
Out: True
```

웹 브라우저로 웹 사이트 접속하기

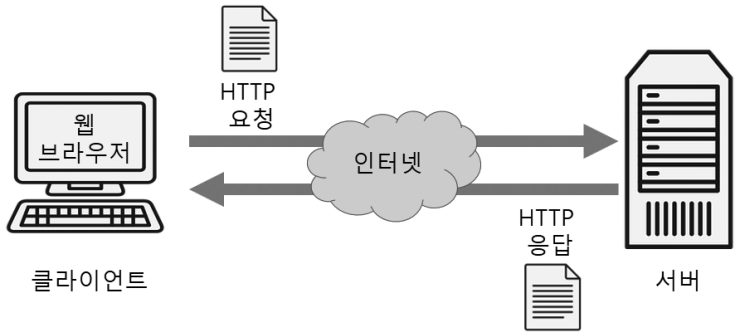
- 여러 개의 웹 사이트에 접속하기

```
In: import webbrowser
    urls = ['www.naver.com', 'www.daum.net', 'www.google.com']
    for url in urls:
        webbrowser.open_new(url)
```

```
In: import webbrowser
   google_url = "www.google.com/#q="
   search_words = ['python web scraping', 'python webbrowser']
   for search_word in search_words:
        webbrowser.open_new(google_url + search_word)
```

웹 스크레이핑을 위한 기본 지식

- 데이터의 요청과 응답 과정

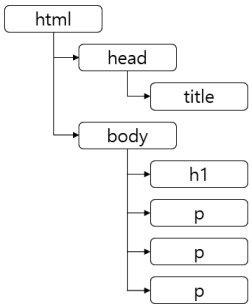


- HTML의 기본 구조

```
<!doctype html>
<html>
 <head>
  <title>이것은 HTML 예제</title>
 </head>
 <body>
  <h1>출간된 책 정보</h1>
  <p id="book_title">이해가 쏙쏙 되는 파이썬</p>
  <p id="author">홍길동</p>
  <p id="publisher">위키북스 출판사</p>
  <p id="year">2018</p>
 </body>
</html>
```

웹 스크레이핑을 위한 기본 지식

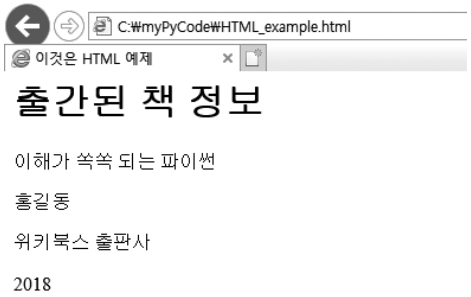
- HTML의 기본 구조




```
In: %%writefile C:₩myPyCode₩HTML_example.html
   <!doctype html>
   <html>
    <head>
     <meta charset="utf-8">
     <title>이것은 HTML 예제</title>
    </head>
    <body>
     <h1>출간된 책 정보</h1>
     <p id="book_title">이해가 쏙쏙 되는 파이썬</p>
     <p id="author">홍길동</p>
     <p id="publisher">위키북스 출판사</p>
     <p id="year">2018</p>
    </body>
   </html>
Out: Writing C:₩myPyCode₩HTML_example.html
```

웹 스크레이핑을 위한 기본 지식

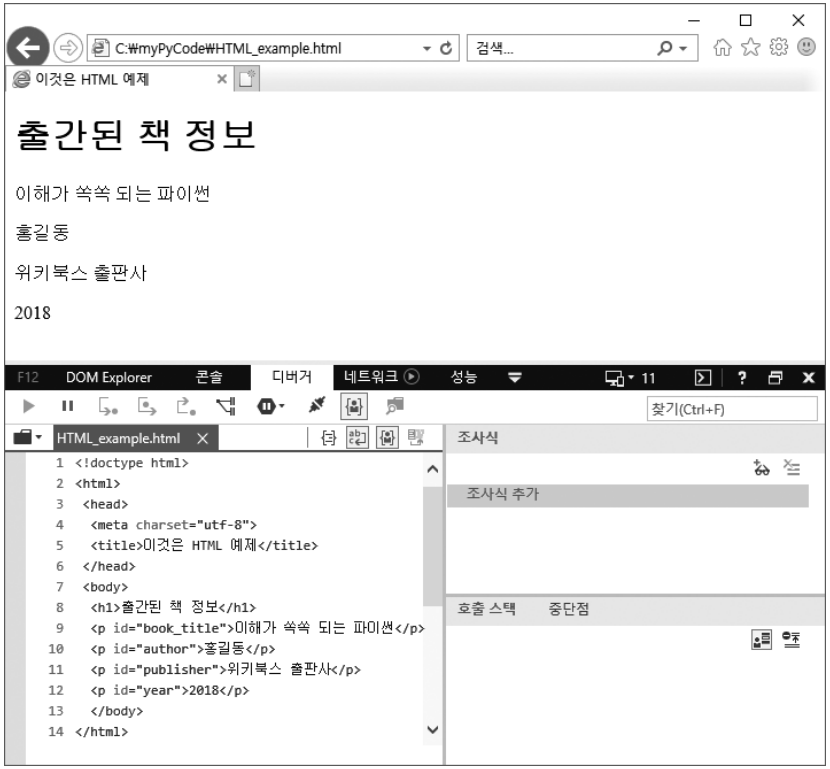
- HTML의 기본 구조




```
In: %%writefile C:/myPyCode/HTML_example2.html
    <!doctype html>
    <html>
     <head>
      <meta charset="utf-8">
      <title>이것은 HTML 예제</title>
     </head>
     <body>
      <h1>출간된 책 정보</h1>
      <p>이해가 쏙쏙 되는 파이썬</p>
      <p>홍길동</p>
      <p>위키북스 출판사</p>
      <p>2018</p>
      </body>
    </html>
Out: Overwriting C:/myPyCode/HTML_example2.html
```

웹 스크레이핑을 위한 기본 지식

- 웹 페이지의 HTML 소스 갖고 오기




```
In: import requests
    r = requests.get("https://www.google.co.kr")
    r
Out: <Response [200]>
```

웹 스크레이핑을 위한 기본 지식

- 웹 페이지의 HTML 소스 갖고 오기

```
In: r.text[0:100]
Out: '<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ko"><head><meta …
```

```
In: import requests
    html = requests.get("https://www.google.co.kr").text
    html[0:100]
Out: '<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ko"><head><meta
     content'
```

웹 스크레이핑을 위한 기본 지식

- HTML 소스코드를 분석하고 처리하기
  - 파싱(Parsing): HTML 코드 구문을 이해하고 요소별로 HTML 코드를 분류
  - Beautiful Soup: 파싱을 돕는 라이브러리
    - https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- 데이터 찾고 추출하기

```
In: from bs4 import BeautifulSoup
    # 테스트용 html 코드
    html = """<html><body><div><span>₩
          <a href=http://www.naver.com>naver</a>₩
          <a href=https://www.google.com>google</a>₩
          <a href=http://www.daum.net/>daum</a>₩
          </span></div></body></html>"""
    # BeautifulSoup를 이용해 HTML 소스를 파싱
    soup = BeautifulSoup(html, 'lxml')
    soup
Out: < html><body><div><span> <a href="http://www.naver.com">naver</a> <a href="https://
     www.google.com">google</a>          <a          href="http://www.daum.net/">daum</a>
</span></div></body></html>
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
In: print(soup.prettify())
Out: <html>
    <body>
     <div>
      <span>
       <a href="http://www.naver.com">
        naver
       </a>
   <a href="https://www.google.com">
        google
       </a>
       <a href="http://www.daum.net/">
        daum
       </a>
      </span>
     </div>
    </body>
   </html>
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
In: soup.find('a')
Out: <a href="http://www.naver.com">naver</a>
```

```
In: soup.find('a').get_text()
Out: 'naver'
```

```
In: soup.find_all('a')
Out: [<a href="http://www.naver.com">naver</a>,
      <a href="https://www.google.com">google</a>,
      <a href="http://www.daum.net/">daum</a>]
```

```
In: site_names = soup.find_all('a')
    for site_name in site_names:
       print(site_name.get_text())
Out: naver
    google
    daum
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
In: from bs4 import BeautifulSoup
    # 테스트용 HTML 코드
    html2 = """
    <html>
    <head>
     <title>작품과 작가 모음</title>
    </head>
    <body>
     <h1>책 정보</h1>
     <p id="book_title">토지</p>
     <p id="author">박경리</p>

     <p id="book_title">태백산맥</p>
     <p id="author">조정래</p>
     <p id="book_title">감옥으로부터의 사색</p>
     <p id="author">신영복</p>
     </body>
   </html>
    """
    soup2 = BeautifulSoup(html2, "lxml")
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
In: soup2.title
Out: <title>작품과 작가 모음</title>
```

```
In: soup2.body
Out: <body>
    <h1>책 정보</h1>
    <p id="book_title">토지</p>
    <p id="author">박경리</p>
    <p id="book_title">태백산맥</p>
    <p id="author">조정래</p>
    <p id="book_title">감옥으로부터의 사색</p>
    <p id="author">신영복</p>
    </body>
```

```
In: soup2.body.h1
Out: <h1>책 정보</h1>
```

```
In: soup2.find_all('p')
Out: [<p id="book_title">토지</p>,
     <p id="author">박경리</p>,
     <p id="book_title">태백산맥</p>,
     <p id="author">조정래</p>,
     <p id="book_title">감옥으로부터의 사색</p>,
     <p id="author">신영복</p>]
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
BeautifulSoup.find_all('태그', '속성')
BeautifulSoup.find('태그', '속성')
```

```
In: soup2.find('p', {"id":"book_title"})
Out: <p id="book_title">토지</p>
```

```
In: soup2.find('p', {"id":"author"})
Out: <p id="author">박경리</p>
```

```
In: soup2.find_all('p', {"id":"book_title"})
Out: [<p id="book_title">토지</p>,
      <p id="book_title">태백산맥</p>,
      <p id="book_title">감옥으로부터의 사색</p>]
```

```
In: soup2.find_all('p', {"id":"author"})
Out: [<p id="author">박경리</p>, <p id="author">조정래</p>, <p id="author">신영복</p>]
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
In: from bs4 import BeautifulSoup

    soup2 = BeautifulSoup(html2, "lxml")

    book_titles = soup2.find_all('p', {"id":"book_title"})
    authors = soup2.find_all('p', {"id":"author"})

    for book_title, author in zip(book_titles, authors):
        print(book_title.get_text() + '/' + author.get_text())

Out: 토지/박경리
     태백산맥/조정래
     감옥으로부터의 사색/신영복
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
In: soup2.select('body h1')
Out: [<h1>책 정보</h1>]
```

```
In: soup2.select('body p')
Out: [<p id="book_title">토지</p>,
      <p id="author">박경리</p>,
      <p id="book_title">태백산맥</p>,
      <p id="author">조정래</p>,
      <p id="book_title">감옥으로부터의 사색</p>,
      <p id="author">신영복</p>]
```

```
In: soup2.select('p')
Out: [<p id="book_title">토지</p>,
      <p id="author">박경리</p>,
      <p id="book_title">태백산맥</p>,
      <p id="author">조정래</p>,
      <p id="book_title">감옥으로부터의 사색</p>,
      <p id="author">신영복</p>]
```

```
In: soup2.select('p#book_title')
Out: [<p id="book_title">토지</p>,
      <p id="book_title">태백산맥</p>,
      <p id="book_title">감옥으로부터의 사색</p>]
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

```
In: soup2.select('p#author')
Out: [<p id="author">박경리</p>, <p id="author">조정래</p>, <p id="author">신영복</p>]
```

```
In: %%writefile C:/myPyCode/HTML_example_my_site.html
  <!doctype html>
  <html>
    <head>
      <meta charset="utf-8">
      <title>사이트 모음</title>
    </head>
    <body>
      <p id="title"><b>자주 가는 사이트 모음</b></p>
      <p id="contents">이곳은 자주 가는 사이트를 모아둔 곳입니다.</p>
      <a href="http://www.naver.com" class="portal" id="naver">네이버</a> <br>
      <a href="https://www.google.com" class="search" id="google">구글</a> <br>
      <a href="http://www.daum.net" class="portal" id="danum">다음</a> <br>
      <a href="http://www.nl.go.kr" class="government" id="nl">국립중앙도서관</a>
    </body>
  </html>
Out: Writing C:/myPyCode/HTML_example_my_site.html
```

웹 스크레이핑을 위한 기본 지식

- 데이터 찾고 추출하기

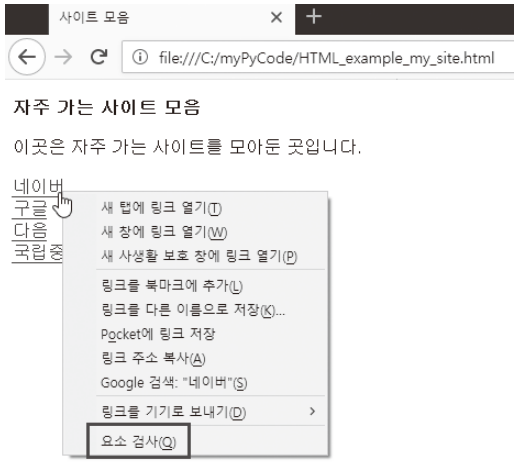
```
In: f = open('C:/myPyCode/HTML_example_my_site.html', encoding='utf-8')

    html3 = f.read()
    f.close()

    soup3 = BeautifulSoup(html3, "lxml")
```

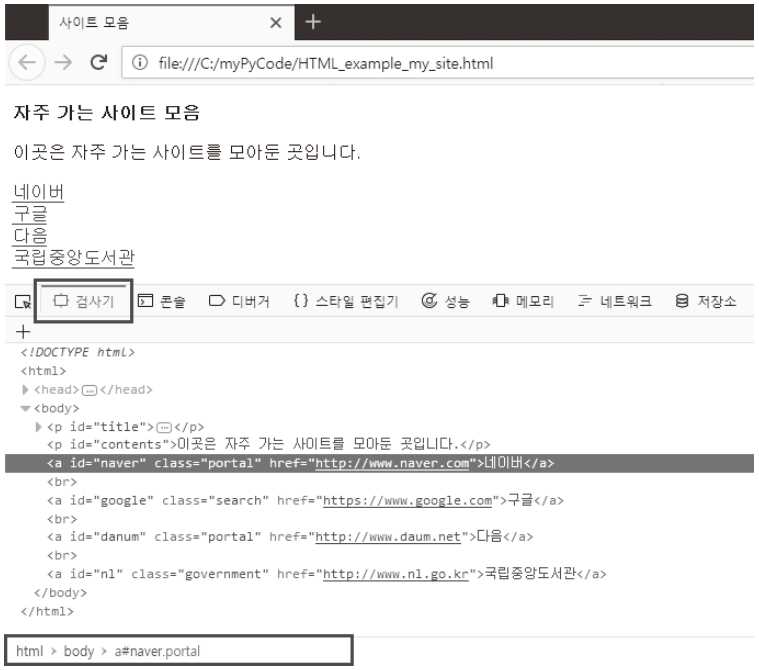
```
In: soup3.select('a')
Out: [<a class="portal" href="http://www.naver.com" id="naver">네이버</a>,
      <a class="search" href="https://www.google.com" id="google">구글</a>,
      <a class="portal" href="http://www.daum.net" id="danum">다음</a>,
      <a class="government" href="http://www.nl.go.kr" id="nl">국립중앙도서관</a>]
```

```
In: soup3.select('a.portal')
Out: [<a class="portal" href="http://www.naver.com" id="naver">네이버</a>,
      <a class="portal" href="http://www.daum.net" id="danum">다음</a>]
```

웹 스크레이핑을 위한 기본 지식

- 웹 브라우저의 요소 검사





웹 스크레이핑을 위한 기본 지식

- 웹 브라우저의 요소 검사

```
soup3.select('html body a')
soup3.select('body a')
soup3.select('html a')
soup3.select('a')
```

```
In: soup3.select('a')
Out: [<a class="portal" href="http://www.naver.com" id="naver">네이버</a>,
      <a class="search" href="https://www.google.com" id="google">구글</a>,
      <a class="portal" href="http://www.daum.net" id="danum">다음</a>,
      <a class="government" href="http://www.nl.go.kr" id="nl">국립중앙도서관</a>]
```

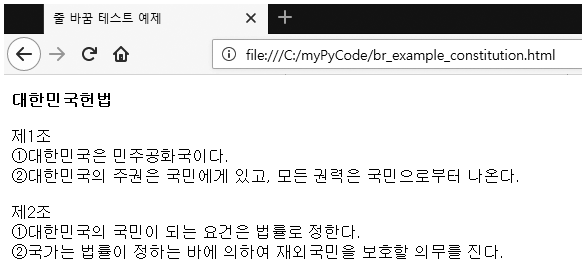
```
In: soup3.select('a.portal')
Out: [<a class="portal" href="http://www.naver.com" id="naver">네이버</a>,
      <a class="portal" href="http://www.daum.net" id="danum">다음</a>]
```

```
In: soup3.select('a#naver')
Out: [<a class="portal" href="http://www.naver.com" id="naver">네이버</a>]
```

웹 스크레이핑을 위한 기본 지식

- 줄 바꿈으로 가독성 높이기

```
In: %%writefile C:/myPyCode/br_example_constitution.html
  <!doctype html>
  <html>
    <head>
      <meta charset="utf-8">
      <title>줄 바꿈 테스트 예제</title>
    </head>
    <body>
    <p id="title"><b>대한민국헌법</b></p>
    <p id="content">제1조 <br/>①대한민국은 민주공화국이다.<br/>②대한민국의 주권은 국민에게 있고,
     모든 권력은 국민으로부터 나온다.</p>
    <p id="content">제2조 <br/>①대한민국의 국민이 되는 요건은 법률로 정한다.<br/>②국가는 법률이
     정하는 바에 의하여 재외국민을 보호할 의무를 진다.</p>
  </body>
</html>
Out: Writing C:/myPyCode/br_example_constitution.html
```



웹 스크레이핑을 위한 기본 지식

- 줄 바꿈으로 가독성 높이기

```
In: from bs4 import BeautifulSoup

    f = open('C:/myPyCode/br_example_constitution.html', encoding='utf-8')

    html_source = f.read()
    f.close()

    soup = BeautifulSoup(html_source, "lxml")

    title = soup.find('p', {"id":"title"})
    contents = soup.find_all('p', {"id":"content"})

    print(title.get_text())
    for content in contents:
        print(content.get_text())

Out: 대한민국헌법
     제1조 ①대한민국은 민주공화국이다.②대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.
     제2조 ①대한민국의 국민이 되는 요건은 법률로 정한다.②국가는 법률이 정하는 바에 의하여 재외국민을 보호할 의무를 진다.
```

웹 스크레이핑을 위한 기본 지식

- 줄 바꿈으로 가독성 높이기

```
find_result = BeautifulSoup.find('태그')
find_result.replace_with('새 태그나 문자열')
```

```
In: html1='<p id="content">제1조<br/>①대한민국은 민주공화국이다.<br/>②대한민국의 주권은 국민에게
          있고, 모든 권력은 국민으로부터 나온다.</p>'
    soup1 = BeautifulSoup(html1, "lxml")
    print("==> 태그 p로 찾은 요소")
    content1 = soup1.find('p', {"id":"content"})
    print(content1)
    br_content = content1.find("br")
    print("==> 결과에서 태그 br로 찾은 요소:", br_content)
    br_content.replace_with("₩n")
    print("==> 태그 br을 개행문자로 바꾼 결과")
    print(content1)
Out: ==> 태그 p로 찾은 요소
     <p id="content">제1조 <br/>①대한민국은 민주공화국이다.<br/>②대한민국의 주권은 국민에게 있고,
    모든 권력은 국민으로부터 나온다.</p>
    ==> 결과에서 태그 br로 찾은 요소: <br/>
    ==> 태그 br을 개행문자로 바꾼 결과
    <p id="content">제1조
     ①대한민국은 민주공화국이다.<br/>②대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터
    나온다.</p>
```

웹 스크레이핑을 위한 기본 지식

- 줄 바꿈으로 가독성 높이기

```
In: soup2 = BeautifulSoup(html1, "lxml")
    content2 = soup2.find('p', {"id":"content"})

    br_contents = content2.find_all("br")
    for br_content in br_contents:
        br_content.replace_with("₩n")
    print(content2)
Out: <p id="content">제1조
    ①대한민국은 민주공화국이다.
    ②대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.</p>
```

```
In: def replace_newline(soup_html):
        br_to_newlines = soup_html.find_all("br")
        for br_to_newline in br_to_newlines:
            br_to_newline.replace_with("₩n")
        return soup_html
```

```
In: soup2 = BeautifulSoup(html1, "lxml")
    content2 = soup2.find('p', {"id":"content"})
    content3 = replace_newline(content2)
    print(content3.get_text())
Out: 제1조
    ①대한민국은 민주공화국이다.
    ②대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.
```

웹 스크레이핑을 위한 기본 지식

- 줄 바꿈으로 가독성 높이기

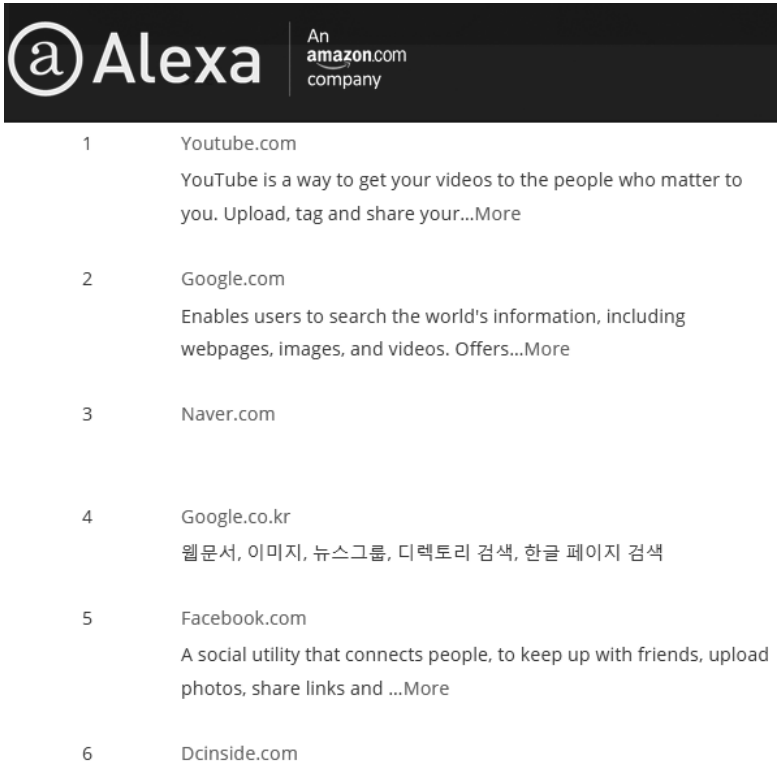
```
In: from bs4 import BeautifulSoup

    soup = BeautifulSoup(html_source, "lxml")

    title = soup.find('p', {"id":"title"})
    contents = soup.find_all('p', {"id":"content"})

    print(title.get_text(), '₩n')

    for content in contents:
        content1 = replace_newline(content)
        print(content1.get_text(),'₩n')

Out: 대한민국헌법
     제1조
     ①대한민국은 민주공화국이다.
     ②대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다.
     제2조
     ①대한민국의 국민이 되는 요건은 법률로 정한다.
     ②국가는 법률이 정하는 바에 의하여 재외국민을 보호할 의무를 진다.
```

웹 사이트에서 데이터 가져오기

- 웹 스크레이핑 시 주의 사항
  - 웹 페이지의 소스코드에서 데이터를 얻기 위한 규칙을 발견할 수 있어야 한다.
  - 파이썬 코드를 이용해 웹 스크레이핑을 할 경우 해당 웹 사이트에 너무 빈번하게 접근하지 말아야 한다.
  - 웹 사이트는 언제든지 예고 없이 변경될 수 있다.
  - 인터넷 상에 공개된 데이터라고 하더라도 저작권이 있는 경우가 있다.

웹 사이트에서 데이터 가져오기

- 순위 데이터를 가져오기
  - 웹사이트 순위
    - https://www.alexa.com/topsites/countries/KR



웹 사이트에서 데이터 가져오기

- 웹사이트 순위

```
In: import requests
    from bs4 import BeautifulSoup

    url = "https://www.alexa.com/topsites/countries/KR"

    html_website_ranking = requests.get(url).text
    soup_website_ranking = BeautifulSoup(html_website_ranking, "lxml")

    # p 태그의 요소 안에서 a 태그의 요소를 찾음
    website_ranking = soup_website_ranking.select('p a')
```

```
In: website_ranking[0:6]

Out: [<a href="/siteinfo/youtube.com">Youtube.com</a>,
     <a href="/siteinfo/google.com">Google.com</a>,
     <a href="/siteinfo/naver.com">Naver.com</a>,
     <a href="/siteinfo/google.co.kr">Google.co.kr</a>,
     <a href="/siteinfo/facebook.com">Facebook.com</a>,
     <a href="/siteinfo/dcinside.com">Dcinside.com</a>]
```

웹 사이트에서 데이터 가져오기

- 웹사이트 순위

```
In: website_ranking[0].get_text()
Out: 'Youtube.com'
```

```
In: website_ranking_address = [website_ranking_element.get_text() for website_ranking_element in website_ranking]
```

```
In: website_ranking_address[0:6]
Out: ['Youtube.com',
      'Google.com',
      'Naver.com',
      'Google.co.kr',
      'Facebook.com',
      'Dcinside.com']
```

웹 사이트에서 데이터 가져오기

– 웹사이트 순위

```
In: import requests
    from bs4 import BeautifulSoup

    url = "https://www.alexa.com/topsites/countries/KR"

    html_website_ranking = requests.get(url).text
    soup_website_ranking = BeautifulSoup(html_website_ranking, "lxml")

    # p 태그의 요소 안에서 a 태그의 요소를 찾음
    website_ranking = soup_website_ranking.select('p a')
    website_ranking_address = [website_ranking_element.get_text() for website_ranking_element in
 website_ranking]

    print("[Top Sites in South Korea]")
    for k in range(6):
        print("{0}: {1}".format(k+1, website_ranking_address[k]))
Out: [Top Sites in South Korea]
     1: Youtube.com
     2: Google.com
     3: Naver.com
     4: Google.co.kr
     5: Facebook.com
     6: Dcinside.com
```

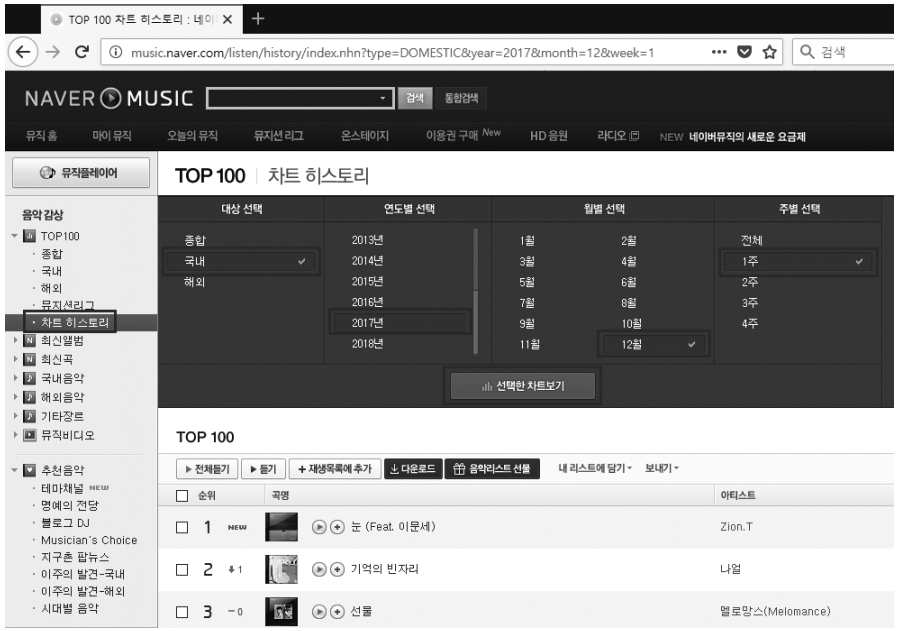
웹 사이트에서 데이터 가져오기

- 웹사이트 순위

```
In: import pandas as pd

    website_ranking_dict = {'Website': website_ranking_address}
     df = pd.DataFrame(website_ranking_dict, columns=['Website'],
    index=range(1,len(website_ranking_address)+1))
    df[0:6]
Out:
          Website
       1  Youtube.com
       2  Google.com
       3  Naver.com
       4  Google.co.kr
       5  Facebook.com
       6  Dcinside.com
```

웹 사이트에서 데이터 가져오기

- 주간 음악 순위
  - http://music.naver.com/listen/history/index.nhn?type=DOMESTIC&year=2017&month=12&week=1



웹 사이트에서 데이터 가져오기

– 주간 음악 순위

```
In: import requests
    from bs4 import BeautifulSoup

    url                                                                                    =
"http://music.naver.com/listen/history/index.nhn?type=TOTAL&year=2017&month=12&week=1"
    html_music = requests.get(url).text
    soup_music = BeautifulSoup(html_music, "lxml")

    # a 태그의 요소 중에서 class 속성값이 "_title" 인 것을 찾고
    # 그 안에서 span 태그의 요소 중에서 class 속성값이 "ellipsis"인 요소를 추출
    titles = soup_music.select('a._title span.ellipsis')
    titles[0:7]
Out: [<span class="ellipsis">눈 (Feat. 이문세)</span>,
     <span class="ellipsis">기억의 빈자리</span>,
     <span class="ellipsis">선물</span>,
     <span class="ellipsis">Beautiful</span>,
     <span class="ellipsis">좋아</span>,
     <span class="ellipsis">피카부 (Peek-A-Boo)</span>,
     <span class="ellipsis">좋니</span>]
```

```
In: music_titles = [title.get_text() for title in titles]
```

```
In: music_titles[0:7]
Out: [ '눈 (Feat. 이문세)', '기억의 빈자리', '선물', 'Beautiful', '좋아', '피카부 (Peek-A-Boo)',
    '좋니']
```

웹 사이트에서 데이터 가져오기

– 주간 음악 순위


```
In: # a 태그의 요소 중에서 class 속성값이 "_artist" 인 것을 찾고
    # 그 안에서 span 태그의 요소 중에서 class 속성값이 "ellipsis"인 요소를 추출
    artists = soup_music.select('a._artist span.ellipsis')
    artists[0].get_text()
Out: '₩r₩n₩t₩t₩t₩r₩n₩t₩t₩t₩r₩n₩t₩t₩tZion.T₩r₩n₩t₩t'
```

웹 사이트에서 데이터 가져오기

- 주간 음악 순위

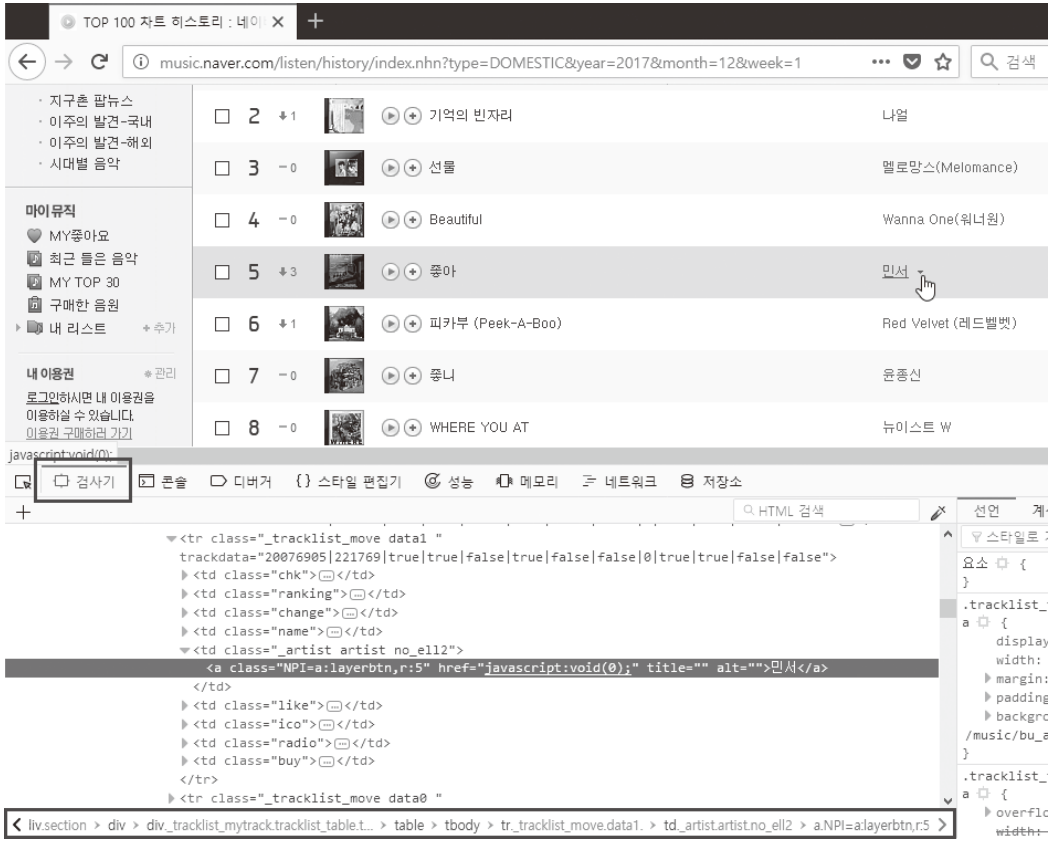
```
In: artists[0].get_text().strip()
Out: 'Zion.T'
```

```
In: music_artists = [artist.get_text().strip() for artist in artists]
```

```
In: music_artists[0:7]
Out: ['Zion.T',
      '나얼',
      '멜로망스(Melomance)',
      'Wanna One(워너원)',
      'Red Velvet (레드벨벳)',
      '윤종신',
      '뉴이스트 W']
```

웹 사이트에서 데이터 가져오기

- 주간 음악 순위



웹 사이트에서 데이터 가져오기

– 주간 음악 순위

```
In: # td 태그의 요소 중에서 class 속성값이 "_artist" 인 것을 찾고
    # 그 안에서 a 태그의 요소를 추출
    artists = soup_music.select('td._artist a')
```

```
In: artists[0]
Out: < a class="_artist NPI=a:artist,r:1,i:115967" href="/artist/home.nhn?artistId=115967"
      title="Zion.T">
     <span class="ellipsis">


              Zion.T
            </span>
    </a>
```

```
In: artists[4]
Out: <a alt="" class="NPI=a:layerbtn,r:5" href="javascript:void(0);" title="">민서</a>
```

```
In: artists[0].get_text().strip()
Out: 'Zion.T'
```

```
In: artists[4].get_text().strip()
Out: '민서'
```

```
In: music_artists = [artist.get_text().strip() for artist in artists]
```

웹 사이트에서 데이터 가져오기

- 주간 음악 순위

```
In: music_artists[0:7]
Out: ['Zion.T',
     '나얼',
     '멜로망스(Melomance)',
     'Wanna One(워너원)',
     '민서',
     'Red Velvet (레드벨벳)',
     '윤종신']
```

웹 사이트에서 데이터 가져오기

– 주간 음악 순위

```
In: import requests
    from bs4 import BeautifulSoup

    url = " http://music.naver.com/listen/history/index.nhn?type=DOMESTIC&year=2017&month=12&week=1&page=1"
    #                                                                                                    url                                                                                                    =
"http://music.naver.com/listen/history/index.nhn?type=DOMESTIC&year=2017&month=12&week=1&page=2"
    # url = "http://music.naver.com/listen/top100.nhn?domain=TOTAL&page=1"

    html_music = requests.get(url).text
    soup_music = BeautifulSoup(html_music, "lxml")

    titles = soup_music.select('a._title span.ellipsis')
    artists = soup_music.select('td._artist a')

    music_titles = [title.get_text() for title in titles]
    music_artists = [artist.get_text().strip() for artist in artists]

    for k in range(7):
      print("{0}: {1} / {2}".format(k+1, music_titles[k], music_artists[k]))
Out: 1: 눈 (Feat. 이문세) / Zion.T
     2: 기억의 빈자리 / 나얼
     3: 선물 / 멜로망스(Melomance)
     4: Beautiful / Wanna One(워너원)
     5: 좋아 / 민서
     6: 피카부 (Peek-A-Boo) / Red Velvet (레드벨벳)
     7: 좋니 / 윤종신
```

웹 사이트에서 데이터 가져오기

– 주간 음악 순위

```
In: import requests
    from bs4 import BeautifulSoup
    import glob

    naver_music_url                                                                                    =
"http://music.naver.com/listen/history/index.nhn?type=DOMESTIC&year=2017&month=12&week=1&pa
ge="

    # 네이버 music 주소를 입력하면 노래 제목과 아티스트를 반환
    def naver_music(url):
        html_music = requests.get(url).text
        soup_music = BeautifulSoup(html_music, "lxml")

        titles = soup_music.select('a._title span.ellipsis')
        artists = soup_music.select('td._artist a')

        music_titles = [title.get_text() for title in titles]
        music_artists = [artist.get_text().strip() for artist in artists]

        return music_titles, music_artists

    # 노래 제목과 아티스트를 저장할 파일 이름을 폴더와 함께 지정
    file_name = 'C:/myPyCode/data/NaverMusicTop100.txt'
```

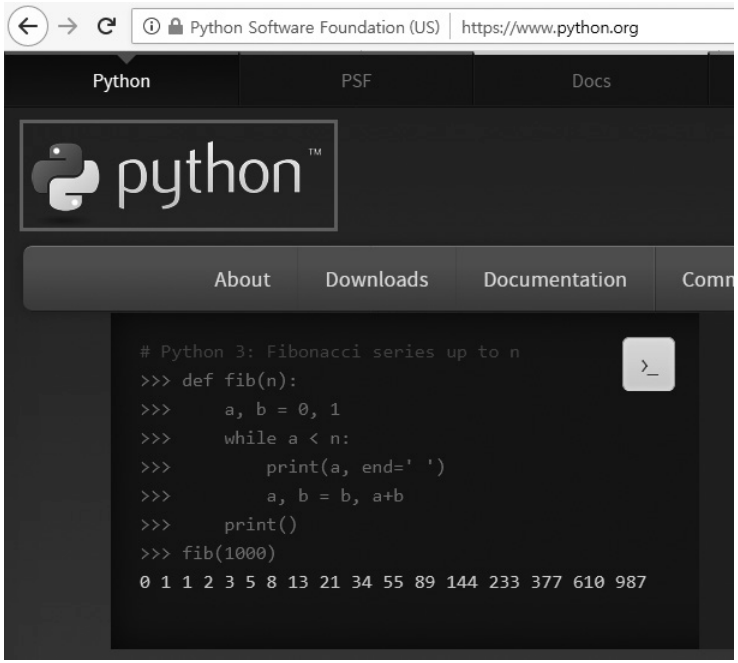
웹 사이트에서 데이터 가져오기

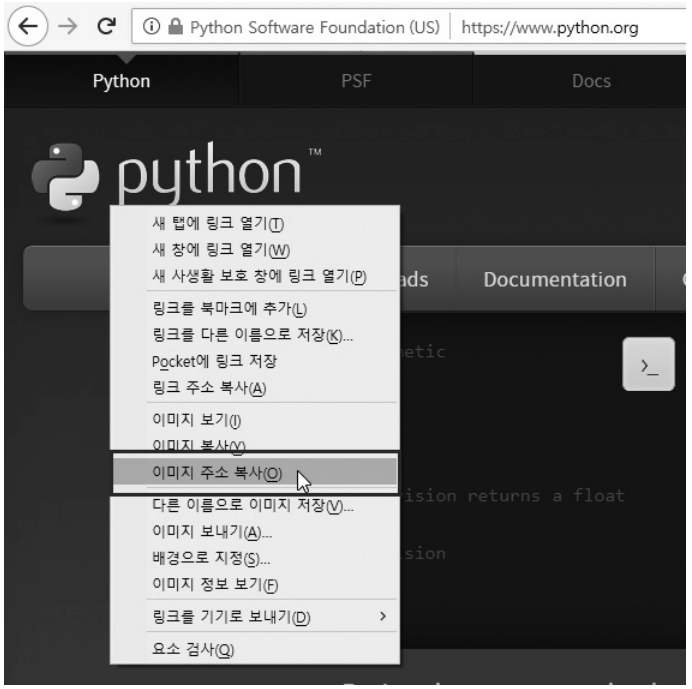
– 주간 음악 순위

```
    f = open(file_name,'w') # 파일 열기

    # 각 page에는 50개의 노래 제목과 아티스트가 추출됨
    for page in range(2):
        naver_music_url_page = naver_music_url + str(page+1) # page URL
        nave_music_titles, naver_music_artists = naver_music(naver_music_url_page)

        # 추출된 노래 제목과 아티스트를 파일에 저장
        for k in range(len(music_titles_artists)):
            f.write("{0:2d}:        {1}/{2}₩n".format(page*50        +        k+1,        nave_music_titles[k],
naver_music_artists[k]))

    f.close() # 파일 닫기

    glob.glob(file_name) # 생성된 파일 확인
Out: ['C:/myPyCode/data/NaverMusicTop100.txt']
```

웹 사이트에서 데이터 가져오기

- 웹 페이지에서 이미지 가져오기
  - https://www.python.org/에 있는 파이썬 로고





https://www.python.org/static/img/python-logo.png

웹 사이트에서 데이터 가져오기

- 웹 페이지에서 이미지 가져오기

```
In: import requests
    url = 'https://www.python.org/static/img/python-logo.png'
    html_image = requests.get(url)
    html_image
Out: <Response [200]>
```

```
In: import os
    image_file_name = os.path.basename(url)
    image_file_name
Out: 'python-logo.png'
```

```
os.makedirs(folder)
```

```
os.path.exists(folder)
```

```
In: folder = 'C:/myPyCode/download'
    if not os.path.exists(folder):
        os.makedirs(folder)
```

```
os.path.join(path1[,path2[,...]])
```

```
In: image_path = os.path.join(folder, image_file_name)
    image_path
Out: 'C:/myPyCode/download₩₩python-logo.png'
```

웹 사이트에서 데이터 가져오기

- 웹 페이지에서 이미지 가져오기

```
In: imageFile = open(image_path, 'wb')
```

```
In: # 이미지 데이터를 100000바이트씩 나눠서 내려받고 파일에 순차적으로 저장
    chunk_size = 1000000
    for chunk in html_image.iter_content(chunk_size):
        imageFile.write(chunk)
    imageFile.close()
```

```
In: os.listdir(folder)
Out: ['python-logo.png']
```

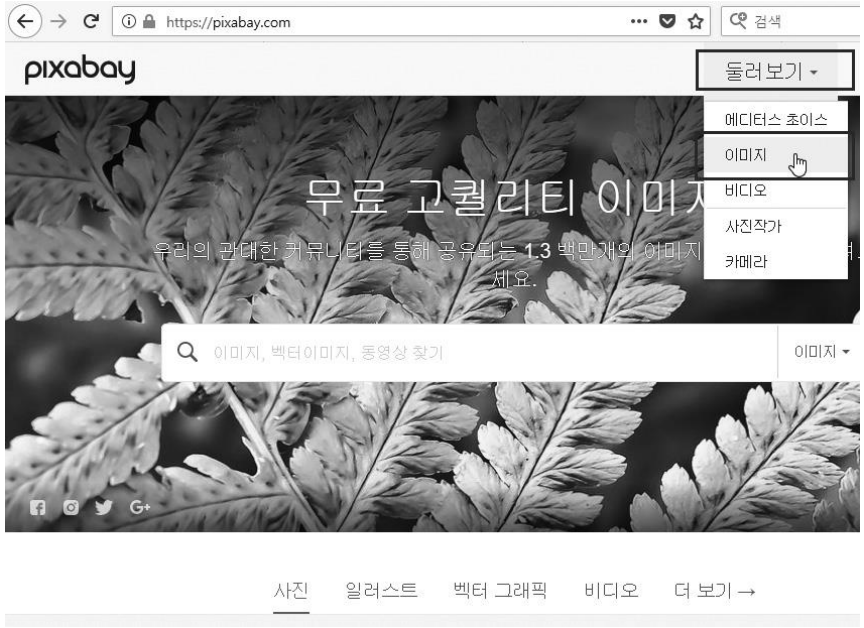
웹 사이트에서 데이터 가져오기

- 웹 페이지에서 이미지 가져오기

```
In: import requests
    import os

    url = 'https://www.python.org/static/img/python-logo.png'
    image_file_name = os.path.basename(url)

    folder = 'C:/myPyCode/download'

    if not os.path.exists(folder):
        os.makedirs(folder)

    image_path = os.path.join(folder, image_file_name)

    imageFile = open(image_path, 'wb')

    # 이미지 데이터를 100000 바이트씩 나눠서 저장
    chunk_size = 1000000
    for chunk in html_image.iter_content(chunk_size):
        imageFile.write(chunk)
    imageFile.close()
```

웹 사이트에서 데이터 가져오기

- 여러 이미지 내려받기
  - https://pixabay.com/ko/photos/?order=popular&cat=animals



웹 사이트에서 데이터 가져오기

- 여러 이미지 내려받기

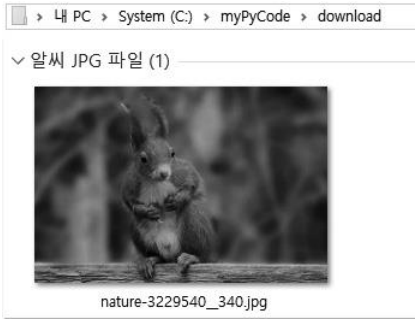
```
In: import requests
    from bs4 import BeautifulSoup

    URL = 'https://pixabay.com/ko/photos/?order=popular&cat=animals'

    html_pixabay_image = requests.get(URL).text
    soup_pixabay_image = BeautifulSoup(html_pixabay_image, "lxml")
    pixabay_image_elements = soup_pixabay_image.select('img')
    pixabay_image_elements[0:3]
Out: [ <img alt="자연, 동물, 다람쥐, 청서 속 주요 Vulgaris의, 설치류, 포유류" src="https://
     cdn.pixabay.com/photo/2018/03/15/21/21/nature-3229540__340.jpg" srcset="https://
     cdn.pixabay.com/photo/2018/03/15/21/21/nature-3229540__340.jpg 1x, https://cdn.pixabay.com/
     photo/2018/03/15/21/21/nature-3229540__480.jpg 2x" title="자연, 동물, 다람쥐, 청서 속 주요
     Vulgaris의, 설치류"/>,
     < img alt="바다, 공석, 모래, 갈매기, 사이드, 봄, 여름, 조류 비행, 하늘" src="https://
      cdn.pixabay.com/photo/2018/03/16/20/13/sea-3232350__340.jpg" srcset="https://
      cdn.pixabay.com/photo/2018/03/16/20/13/sea-3232350__340.jpg 1x, https://cdn.pixabay.com/
      photo/2018/03/16/20/13/sea-3232350__480.jpg 2x" title="바다, 공석, 모래, 갈매기, 사이드, 봄,
      여름, 조류 비행"/>,
     < img alt="공상, 사슴, 포유 동물, 숲, 자연, 야외 활동, 빛, 햇빛, 잔디" src="https://
      cdn.pixabay.com/photo/2018/03/16/21/24/fantasy-3232570__340.jpg" srcset="https://
      cdn.pixabay.com/photo/2018/03/16/21/24/fantasy-3232570__340.jpg 1x, https://
      cdn.pixabay.com/photo/2018/03/16/21/24/fantasy-3232570__480.jpg 2x" title="공상, 사슴, 포유
      동물, 숲, 자연, 야외 활동, 빛, 햇빛"/>]
```

웹 사이트에서 데이터 가져오기

– 여러 이미지 내려받기

```
In: pixabay_image_url = pixabay_image_elements[0].get('src')
    pixabay_image_url
Out: 'https://cdn.pixabay.com/photo/2018/03/15/21/21/nature-3229540__340.jpg'
```

```
In: html_image = requests.get(pixabay_image_url)

    folder = "C:/myPyCode/download"

    # os.path.basename(URL)는 웹사이트나 폴더가 포함된 파일명에서 파일명만 분리하는 방법
    imageFile = open(os.path.join(folder, os.path.basename(pixabay_image_url)), 'wb')

    # 이미지 데이터를 100000 바이트씩 나눠서 저장하는 방법
    chunk_size = 1000000
    for chunk in html_image.iter_content(chunk_size):
        imageFile.write(chunk)
    imageFile.close()
```



웹 사이트에서 데이터 가져오기

- 여러 이미지 내려받기

```
In: import requests
    from bs4 import BeautifulSoup
    import os
    # URL(주소)에서 이미지 주소 추출
    def get_image_url(url):
        html_image_url = requests.get(url).text
        soup_image_url = BeautifulSoup(html_image_url, "lxml")
        image_elements = soup_image_url.select('img')
        if(image_elements != None):
            image_urls = []
            for image_element in image_elements:
                image_urls.append(image_element.get('src'))
            return image_urls
        else:
            return None

    # 폴더를 지정해 이미지 주소에서 이미지 내려받기
    def download_image(img_folder, img_url):
        if(img_url != None):
            html_image = requests.get(img_url)
            # os.path.basename(URL)는 웹사이트나 폴더가 포함된 파일명에서 파일명만 분리
            imageFile = open(os.path.join(img_folder, os.path.basename(img_url)), 'wb')
            chunk_size = 1000000 # 이미지 데이터를 100000바이트씩 나눠서 저장
```

웹 사이트에서 데이터 가져오기

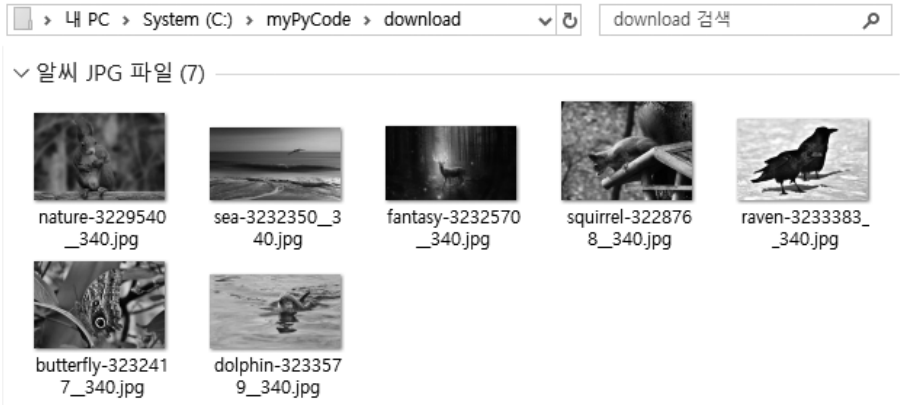
- 여러 이미지 내려받기

```
        for chunk in html_image.iter_content(chunk_size):
            imageFile.write(chunk)
            imageFile.close()
        print("이미지 파일명: '{0}'. 내려받기 완료!".format(os.path.basename(img_url)))
    else:
        print("내려받을 이미지가 없습니다.")

# 웹 사이트의 주소 지정
pixabay_url = 'https://pixabay.com/ko/photos/?order=popular&cat=animals'
# pixabay_url= 'https://pixabay.com/ko/photos/?order=popular&cat=animals&pagi=2'

figure_folder = "C:/myPyCode/download" # 이미지를 내려받을 폴더 지정

pixabay_image_urls = get_image_url(pixabay_url) # 이미지 파일의 주소 가져오기

num_of_download_image = 7 # 내려받을 이미지 개수 지정
# num_of_download_image = len(pixabay_image_urls)

for k in range(num_of_download_image):
    download_image(figure_folder,pixabay_image_urls[k])
print("================================")
print("선택한 모든 이미지 내려받기 완료!")
```

웹 사이트에서 데이터 가져오기

– 여러 이미지 내려받기

```
Out: 이미지 파일명: 'nature-3229540__340.jpg'. 내려받기 완료!
     이미지 파일명: 'sea-3232350__340.jpg'. 내려받기 완료!
     이미지 파일명: 'fantasy-3232570__340.jpg'. 내려받기 완료!
     이미지 파일명: 'raven-3233383__340.jpg'. 내려받기 완료!
     이미지 파일명: 'squirrel-3228768__340.jpg'. 내려받기 완료!
     이미지 파일명: 'butterfly-3232417__340.jpg'. 내려받기 완료!
     이미지 파일명: 'dolphin-3233579__340.jpg'. 내려받기 완료!
     ==================================
     선택한 모든 이미지 내려받기 완료!
```




```
In: num_of_download_image = len(pixabay_image_urls)
num_of_download_image
Out: 100
```

감사합니다.

※ 본 교안은 강의 수강 용도로만 사용 가능합니다.
상업적 이용을 일절 금함.

사람투자 과학기술정보통신부 nipa 정보통신산업진흥원 National IT Industry Promotion Agency KSA 한국표준협회