Lab. Using Beautiful Soup

1. BeautifulSoup module 이용하기
   1)pip install BeautifulSoup4
   2)Scraping을 위한 module
   3)루이스 캐럴의 [이상한 나라의 앨리스]에 나오는 동명의 시에서 이름을 따왔다.
   4)일반적으로 HTML tag들이 start tag와 end tag가 서로 pair 되지 않을 경우가 많다.
   5)pair 되지 않아도 아름답게 처리해 주는 module
   6)https://www.crummy.com/software/BeautifulSoup/bs4/doc/

```
from bs4 import BeautifulSoup
import requests

html_data = requests.get('https://www.naver.com')
soup = BeautifulSoup(html_data.text, 'html.parser')
soup.title
---------------------------------
<title>NAVER</title>
```

   7)첫번째 인자는 Python 객체로 바꿀 string을 넣어주고 두 번째 인자로 parser를 넣는다.
   8)Parser란, 원시 코드인 순수 문자열 객체를 해석할 수 있도록 분석하는 것을 의미한다.
   9)Python에서 사용되는 parser는 다음과 같다.
      a. lxml
      b. html5lib
      c. html.parser

   10)각 parser의 장단점을 알아보자.
      a. lxml
         -XML 해석이 가능한 parser이다.
         -Python 2.x와 3.x 모두 지원 가능하다.
         -다른 parser에 비해 매우 빠른 속도로 처리한다.
         -그 이유는 C언어로 구현되어 있기 때문이다.
      b. html5lib
         -웹 브라우저 방식으로 HTML을 해석한다.
         -하지만 처리 속도가 매우 느리다는 단점이 있다.
         -그리고 2.x 전용이다.
      c. html.parser
         -최신 버전의 Python에서는 사용이 불가

2. 각 parser 비교하기
   1)lxml

```
from bs4 import BeautifulSoup
html = """<p>test</p>"""
soup = BeautifulSoup(html, 'lxml')
print(soup)
-----------------------------------------------
<html><body><p>test</p></body></html>


from bs4 import BeautifulSoup
html = """<html><p>test</p></html>"""
soup = BeautifulSoup(html, 'lxml')
print(soup)

html = """<body><p>test</p></body>"""
soup = BeautifulSoup(html, 'lxml')
print(soup)
-------------------------------------------
<html><body><p>test</p></body></html>
<html><body><p>test</p></body></html>
```

      -html과 body tag가 포함된 형태로 만들어 준다.

   2)html5lib

```
$ pip install html5lib

from bs4 import BeautifulSoup
html = """<p>test</p>"""
soup = BeautifulSoup(html, 'html5lib')
print(soup)
-------------------------------------------------
<html><head></head><body><p>test</p></body></html>
```

      -html5lib도 html처럼 해석하기 때문에 html, head, body tag가 포함된 형태로 만들어 준다.

2. 한빛미디어 책 제목 읽어오기

```
hanbit = requests.get('http://www.hanbit.co.kr/media/')
soup = BeautifulSoup(hanbit.text, 'html.parser')
soup
```

```
    ---------------------------------------------------------------
    <!DOCTYPE html>
    <html lang="ko">
    <head>
    <!--[if lte IE 8]>
    <script>
      location.replace('/support/explorer_upgrade.html');
    </script>
    <![endif]-->
    <!-- Google Tag Manager -->
    <script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
    new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
    j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
    'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
    })(window,document,'script','dataLayer','GTM-W9D5PM3');</script>
    <!-- End Google Tag Manager -->
    ...
    ...

    for book in soup.find_all('p', class_='book_tit'):
        print(book.find('a').text)
    ---------------------------------------------------------------
    리얼월드 HTTP : 역사와 코드로 배우는 인터넷과 웹 기술
    이것이 MariaDB다
    제프리 리처의 Windows via C/C++(복간판)
    초보자를 위한 유니티 입문(개정판) : 따라 하면서 배우는 2D & 3D 게임 개발
    회사에서 바로 통하는 실무 엑셀
    맛있는 디자인 포토샵 CC 2019
    알고리즘이 욕망하는 것들
    파이썬 라이브러리를 활용한 머신러닝(번역개정판) : 사이킷런 핵심 개발자가 쓴 머신러닝과 데이터 과학 실무서
    파이썬으로 웹 크롤러 만들기(2판) : 초간단 나만의 웹 크롤러로 원하는 데이터를 가져오는 방법
    더 나은 세상을 위한 소프트 디지털
    비됴클래스 하줜의 유튜브 동영상 편집 with 프리미어 프로
    회사에서 바로 통하는 실무 엑셀+파워포인트+워드&한글
    맛있는 디자인 포토샵&일러스트레이터 CC 2019
    맛있는 디자인 프리미어 프로&애프터 이펙트 CC 2019
    밑바닥부터 시작하는 딥러닝
    이것이 C#이다
    핸즈온 머신러닝
    소문난 명강의 : 레트로의 유니티 게임 프로그래밍 에센스
    이것이 자바다
    이것이 우분투 리눅스다


3. Naver 영화 평점 Scraping 하기

    from bs4 import BeautifulSoup

    html_data = requests.get('https://movie.naver.com/movie/point/af/list.nhn?page=1')
    soup = BeautifulSoup(html_data.text, 'html.parser')
    titles = soup.find_all(class_='movie')

    title_list = []
    for title in titles:
        print(title.text)
    -------------------------------------------
    대학살의 신
    스타워즈: 라스트 제다이
    내안의 그놈
    말모이
    말모이
    내안의 그놈
    언니
    내안의 그놈
    존 윅
    마이 리틀 자이언트

    for title in titles:
        title_list.append(title.text)

    point_list = []
    points = soup.find_all(class_='point')
    for point in points:
        point_list.append(point.text)

    review_list = []
    reviews = soup.find_all(class_='title')
    for review in reviews:
         rev = review.text
         rev = rev.strip()
         rev = rev.replace('\t', '')
         rev = rev.replace('\n', '')
         rev = rev.replace('신고', '')
```

```
        review_list.append(rev)

    df = pd.DataFrame(title_list, columns=['Title'])
    df['Point'] = point_list
    df['Review'] = review_list
    df
    ------------------------------------------------
       Title                        Point     Review
    0  대학살의 신                   7           대학살의 신자식싸움에 부모등 터진다
    1  스타워즈: 라스트 제다이      1          스타워즈: 라스트 제다이이거 보느니 로그원 열번 보는게 낫다
    2  내안의 그놈                   10          내안의 그놈재밌어요 유치해도 빵빵터짐 진영 연기 잘하네요 라미란과 잘...
    3  말모이                      10        말모이후반부에 보고 울었습니다 감동적임
    4  말모이                      10        말모이재미를 떠나서 역사는 무조건 10점이다
    5  내안의 그놈                   10          내안의 그놈뻔한스토리이지만 재밌게 잘 보고왔어요최고의성형은 다이어트
    6  언니                         5           언니론다 로우지가 언니 역활했으면 그나마 공감이 됐을듯...개연성도 떨어지고 액션도...
    7  내안의 그놈                   9           내안의 그놈ㅋㅋ재밌었음 생각보다 안정적인 연기력 소소하게 웃기 좋은 영화
    8  존 윅                        10          존 윅액션의 선두주자 키아누리브스!!
    9  마이 리틀 자이언트           7           마이 리틀 자이언트동화보다 더 환상적인 거인


4. Naver 평점 1page부터 100page까지 scraping 하기

    from bs4 import BeautifulSoup
    import requests
    import pandas as pd

    url = 'https://movie.naver.com/movie/point/af/list.nhn?page='

    title_list = []
    point_list = []
    review_list = []

    for pge in range(1, 101):
        url = url + str(pge)
        print(url)
        html_data = requests.get(url)
        soup = BeautifulSoup(html_data.text, 'html.parser')
        titles = soup.find_all(class_='movie')
        points = soup.find_all(class_='point')
        reviews = soup.find_all(class_='title')
        for title in titles:
            title_list.append(title.text)
        for point in points:
            point_list.append(point.text)
        for review in reviews:
            rev = review.text
            rev = rev.strip()
            rev = rev.replace('\t', '')
            rev = rev.replace('\n', '')
            rev = rev.replace('신고', '')
            review_list.append(rev)
        url = url.split('=')[0] + '='

    df = pd.DataFrame(title_list, columns=['Title'])
    df['Point'] = point_list
    df['Review'] = review_list

    df.info()
    ------------------------------------------------
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 1020 entries, 0 to 1019
    Data columns (total 3 columns):
    Title     1020 non-null object
    Point     1020 non-null object
    Review    1020 non-null object
    dtypes: object(3)
    memory usage: 24.0+ KB


5. Coupang의 상품정보 Scraping
    1)Web 문서들은 서로 다양한 문서 구조로 출력된다.
    2)따라서 Python에서 web 문서로부터 scraping을 하기 위해서는 추출하고자 하는 정보들이 구성되어 있는 영역을 먼저 확인해야 한다.
    3)Social Commerce의 대표적인 online market인 Coupang의 상품 정보 추출을 해보자.
    4)'여성패션' 중 '여성 크로스백' 목록 item을 살펴보자.
    5)Scraping하려는 web page의 URL 구조와 문서 구조를 파악한다.
    6)URL 구조
        http://www.coupang.com/np/search?q=여성크로스백
    7)문서 구조
      -상품명 : class="name"
      -가격 : class="price-value"
```

6. 한국일보 headline 기사 Scraping하기
   1)한국일보 첫 page의 기사를 Scraping 해보자.
   2)먼저 scraping 하려는 web page의 URL 구조와 문서 구조를 파악해야 한다.
   3)URL 구조
      http://www.hankookilbo.com/
   4)문서 구조
      -기사 제목 : class=”title”