

파이썬 Beautiful Soup



파이썬 Beautiful Soup

- **Beautiful Soup**
 - HTML 및 XML 문서 를 구문 분석하기 위한 Python 패키지
- **Beautiful Soup 객체 생성**
 - 인스턴스명 = BeautifulSoup(markup, parser)
- **Beautiful Soup 파서 종류**
 - **html.parser** : 파이썬 표준 라이브러리에 포함된 파서로 빠르지만 유연하지 않기 때문에 단순한 HTML문서에 사용
 - BeautifulSoup(markup, "html.parser")
 - **lxml** : 매우 빠르고 유연하지만 외부 의존
 - lxml의 HTML 파서
 - BeautifulSoup(markup, "lxml")
 - lxml의 XML 파서
 - BeautifulSoup(markup, "lxml-xml")
 - **html5lib** : 웹 브라우저와 동일한 방식으로 페이지 구문 분석하는 파서로 매우 느리지만 매우 유연하지만 외부 의존
 - BeautifulSoup(markup, "html5lib")

Beautiful Soup 태그 파싱

- `.find(태그명)`
 - 조건에 맞는 태그 1개만 찾음
- `.find_all(태그명)`
 - 조건에 맞는 모든 태그 찾음

```
bs = BeautifulSoup(html, "html.parser")
```

#태그 가져오기

```
body = bs.body
```

```
li = body.li
```

```
print(li)
```

```
print(type(li))
```

```
print("-"*20)
```

```
li = body.find("li")
```

```
print(li)
```

```
print(type(li))
```

```
print("-"*20)
```

```
li = body.find_all("li")
```

```
print(li)
```

```
print(type(li))
```

```
print("-"*20)
```

```
<li><a href="#m1">원칙1</a></li>  
<class 'bs4.element.Tag'>
```

```
-----  
<li><a href="#m1">원칙1</a></li>  
<class 'bs4.element.Tag'>
```

```
-----  
[<li><a href="#m1">원칙1</a></li>, <li><a href="#m2">원칙2</a></li>, <li><a href="#m3">원칙3  
구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유</li>, <li class="c1" id="m2">소프트웨어  
유</li>, <li class="c2" id="m3">소프트웨어를 향상시키고 이를 공동체 전체의 이익을 위해서 다스  
<class 'bs4.element.ResultSet'>
```



Beautiful Soup 태그 파싱

```
lis = body.find_all("li")
print(lis)
print(type(lis))
print("-"*20)
```

```
for li in lis :
    print(li)
print("-"*20)
```

```
for li in lis :
    if li.find("a") : print(li.find("a"))
print("-"*20)
```

```
for li in lis :
    if li.find("a") : print(li.find("a").text)
print("-"*20)
```

```
-----
<li><a href="#m1">원칙1</a></li>
<li><a href="#m2">원칙2</a></li>
<li><a href="#m3">원칙3</a></li>
<li class="c1" id="m1">소프트웨어의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유</li>
<li class="c1" id="m2">소프트웨어를 이웃과 함께 공유하기 위해서 이를 복제하고 배포할 수 있는 자유</li>
<li class="c2" id="m3">소프트웨어를 향상시키고 이를 공동체 전체의 이익을 위해서 다신 환원시킬 수 있는 자유</li>
```

```
-----
<a href="#m1">원칙1</a>
<a href="#m2">원칙2</a>
<a href="#m3">원칙3</a>
```

```
-----
원칙1
원칙2
원칙3
-----
```



공공데이터

- <https://www.data.go.kr>

「공공데이터의 제공 및 이용 활성화에 관한 법률」 제21조(공공데이터포털의 운영)

제21조(공공데이터포털의 운영) ① 행정안전부장관은 공공데이터의 효율적 제공을 위하여 통합제공시스템(이하 "공공데이터포털"이라 한다)을 구축·관리하고 활용을 촉진하여야 한다.

② 행정안전부장관은 공공기관의 장에게 공공데이터포털의 구축과 운영에 필요한 공공데이터의 연계, 제공 등의 협력을 요청할 수 있다. 이 경우 요청을 받은 공공기관의 장은 특별한 사유가 없는 한 이에 따라야 한다.

③ 그 밖에 공공데이터포털의 구축·관리 및 활용촉진 등 필요한 사항은 대통령령으로 정한다.








예제 공공데이터

- <http://www.kobis.or.kr/kobisopenapi/homepg/apiservice/searchServiceInfo.do>

제공 서비스

영화관입장권통합전산망이 제공하는 오픈API 서비스 모음입니다.
사용 가능한 서비스를 확인하고 서비스별 인터페이스 정보를 조회합니다.

 1 박스오피스	<ul style="list-style-type: none"> • 일별 박스오피스 • 주간/주말 박스오피스
 2 공통코드조회	<ul style="list-style-type: none"> • 공통코드 조회
 3 영화정보	<ul style="list-style-type: none"> • 영화목록 • 영화 상세 정보
 4 영화사정보	<ul style="list-style-type: none"> • 영화사 목록 • 영화사 상세 정보
 5 영화인정보	<ul style="list-style-type: none"> • 영화인 목록 • 영화인 상세 정보

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<boxOfficeResult>
  <boxOfficeType>일별 박스오피스</boxOfficeType>
  <showRange>20120101~20120101</showRange>
  <dailyBoxOfficeList>
    <dailyBoxOffice>
      <num>1</num>
      <rank>1</rank>
      <rankInten>0</rankInten>
      <rankOldAndNew>OLD</rankOldAndNew>
      <movieCd>20112207</movieCd>
      <movieNm>미션임파서블:고스트프로토콜</movieNm>
      <openDt>2011-12-15</openDt>
      <salesAmt>2776060500</salesAmt>
      <salesShare>36.3</salesShare>
      <salesInten>415699000</salesInten>
      <salesChange>-13</salesChange>
      <salesAcc>40541108500</salesAcc>
      <audiCnt>353274</audiCnt>
      <audiInten>-60106</audiInten>
      <audiChange>-14.5</audiChange>
      <audiAcc>5328435</audiAcc>
      <scrnCnt>697</scrnCnt>
      <showCnt>3223</showCnt>
    </dailyBoxOffice>
  </dailyBoxOfficeList>
  <dailyBoxOffice>
    <num>2</num>
    <rank>2</rank>
    <rankInten>1</rankInten>
    <rankOldAndNew>OLD</rankOldAndNew>
    <movieCd>20110295</movieCd>
    <movieNm>마이 웨이</movieNm>
    <openDt>2011-12-21</openDt>
    <salesAmt>1189058500</salesAmt>
    <salesShare>15.6</salesShare>
    <salesInten>-105894500</salesInten>
    <salesChange>-8.2</salesChange>
    <salesAcc>13002897500</salesAcc>
    <audiCnt>153501</audiCnt>
    <audiInten>-16465</audiInten>
    <audiChange>-9.7</audiChange>
    <audiAcc>1739543</audiAcc>
    <scrnCnt>588</scrnCnt>
    <showCnt>2321</showCnt>
  </dailyBoxOffice>
  <dailyBoxOffice>
    <num>3</num>
    <rank>3</rank>
    <rankInten>-1</rankInten>
    <rankOldAndNew>OLD</rankOldAndNew>
    <movieCd>20112621</movieCd>
    <movieNm>설류공주:그림자 게임</movieNm>
    <openDt>2011-12-21</openDt>
    <salesAmt>1176022500</salesAmt>
    <salesShare>15.4</salesShare>
    <salesInten>-210328500</salesInten>
    <salesChange>-15.2</salesChange>
    <salesAcc>10678327500</salesAcc>
    <audiCnt>153004</audiCnt>
    <audiInten>-31283</audiInten>
    <audiChange>-17</audiChange>
    <audiAcc>1442861</audiAcc>
```

예제 공공데이터

- https://www.weather.go.kr/weather/lifenindustry/sevice_rss.jsp

The screenshot shows the 'RSS' tab selected in the top navigation bar. Below it, there is a section titled 'RSS란?' (What is RSS?) explaining the concept. Another section titled 'RSS 서비스 이용하기' (How to use RSS service) contains a flowchart with four steps: 'RSS리더기 설치' (Install RSS reader), '구독을 원하는 정보의 RSS주소 복사' (Copy RSS address of information you want to subscribe), '복사된 RSS주소를 RSS리더기에 추가' (Add copied RSS address to RSS reader), and 'RSS리더기를 통해 실시간으로 정보를 확인' (Check information in real-time through RSS reader). At the bottom, there is a search bar for '동네예보' (Local forecast) with a dropdown menu set to '서울특별시' (Seoul) and a search button. An 'RSS' button is also visible.

<http://www.kma.go.kr/wid/queryDFSRSS.jsp?zone=2641060000>

The screenshot shows a web browser displaying the RSS feed content. The address bar shows the URL: www.kma.go.kr/wid/queryDFSRSS.jsp?zone=2641060000. The page content shows the XML feed structure, including the channel title '기상청 동네예보 웹서비스 - 부산광역시 금정구 장전제1동 도표예보' and the item title '동네예보(도표) : 부산광역시 금정구 장전제1동 [X=98,Y=77]'. The item description contains the URL 'http://www.kma.go.kr/weather/forecast/timeseries.jsp?searchType=INTEREST&dongCode=2641060000'. The item's header and body sections are also visible, showing the time '202001291100' and the temperature '15'.

XML 데이터 크롤링

- BeautifulSoup을 이용한 파싱

```
from bs4 import BeautifulSoup
def dailyBoxOfficeList(data) :
    bs = BeautifulSoup(data, 'lxml-xml')
    dailyBoxOffices = bs.find_all("dailyBoxOffice")

    return dailyBoxOffices
```



웹 크롤링

- 스크래핑(scraping)

- HTTP를 통해 웹 사이트의 내용을 긁어다 원하는 형태로 가공하는 것
- 웹 사이트의 데이터를 수집하는 모든 작업

- 크롤링(crawling)

- 여러 인터넷 사이트의 페이지(문서, html 등)를 수집해서 분류하는 것
- 크롤러는 조직적, 자동화된 방법으로 웹을 탐색하는 프로그램으로 크롤러가 하는 작업을 크롤링이라고 함

- 파싱(parsing)

- 웹 페이지에서 원하는 데이터를 특정 패턴이나 순서로 추출하여 정보를 가공하는 것

- 웹 크롤링

- 인터넷 상에 존재하는 자료를 스크래핑(크롤링)을 통해 수집하여 데이터를 파싱하여 원하는 정보를 추출하는 것



Beautiful Soup CSS선택자 파싱

- `.select_one(선택자)`, `.select(선택자)`

`#select` 가져오기

```
m1 = body.select_one("#m1")
print(m1)
print("-"*20)
```

```
c1 = body.select_one(".c1")
print(c1)
print("-"*20)
```

```
c1 = body.select(".c1")
print(c1)
print("-"*20)
```

```
hrefs = body.select("a[href]")
for href in hrefs:
    print(href)
print("-"*20)
```

```
hrefs = body.select("ul > li > a")
for href in hrefs:
    print(href.text)
print("-"*20)
```

```
<li class="c1" id="m1">소프트웨어의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유</li>
-----
<li class="c1" id="m1">소프트웨어의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유</li>
-----
[<li class="c1" id="m1">소프트웨어의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유</li>,
 함께 공유하기 위해서 이를 독제하고 배포할 수 있는 자유</li>]
-----
<a href="#m1">원칙1</a>
<a href="#m2">원칙2</a>
<a href="#m3">원칙3</a>
<a href="https://www.fsf.org/">

</a>
-----
원칙1
원칙2
원칙3
-----
```



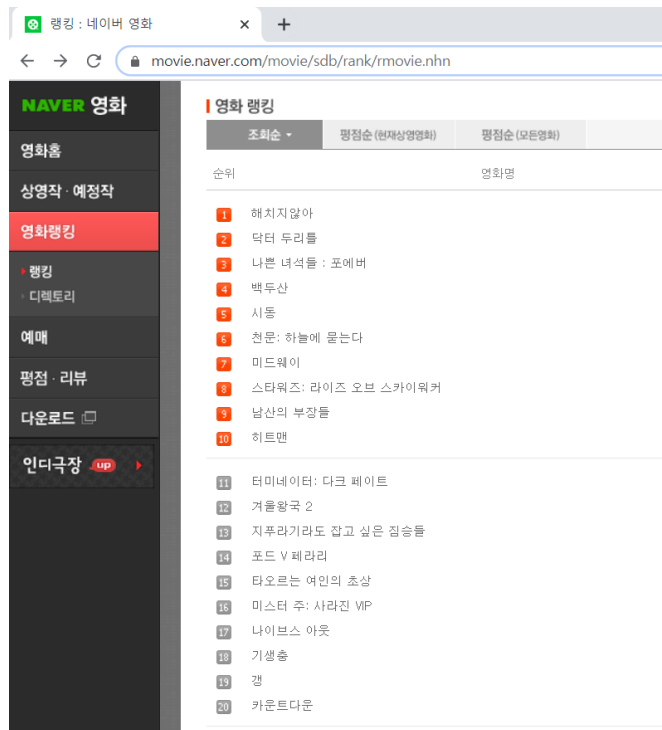
실습

• 그림의 위치를 추출하시오.



실습

- 네이버 영화 사이트에서 영화 순위를 화면에 표시하시오.



- 1 위: 해치지않아
- 2 위: 닥터 두리틀
- 3 위: 나쁜 녀석들 : 포에버
- 4 위: 백두산
- 5 위: 시동
- 6 위: 천문: 하늘에 묻는다
- 7 위: 미드웨이
- 8 위: 스타워즈: 라이즈 오브 스카이워커
- 9 위: 남산의 부장들
- 10 위: 히트맨
- 11 위: 터미네이터: 다크 페이트
- 12 위: 겨울왕국 2
- 13 위: 지푸라기라도 잡고 싶은 짐승들
- 14 위: 포드 V 페라리
- 15 위: 타오르는 여인의 초상
- 16 위: 미스터 주: 사라진 VIP
- 17 위: 나이트스 아웃
- 18 위: 기생충
- 19 위: 갯
- 20 위: 카운트다운
- 21 위: 라스트 선라이즈
- 22 위: 극장판 원피스 스탬피드
- 23 위: 눈의 여왕4
- 24 위: 피아니스트의 전설
- 25 위: 인셉션
- 26 위: 신비아파트: 극장판 하늘도깨비 대 요르문간드

실습

- 다음 영화 사이트에서 입력년도에서 출력년도까지 자료를 추출하시오.

박스오피스 | 다음영화

movie.daum.net/boxoffice/yearly?year=2019

8 김병민 95

로그인

영화 연예

통합 검색

홈 현재상영/개봉예정 박스오피스 박른예매 뉴스

내평점 예매내역

주간 월간 **연간**

< 2019 >

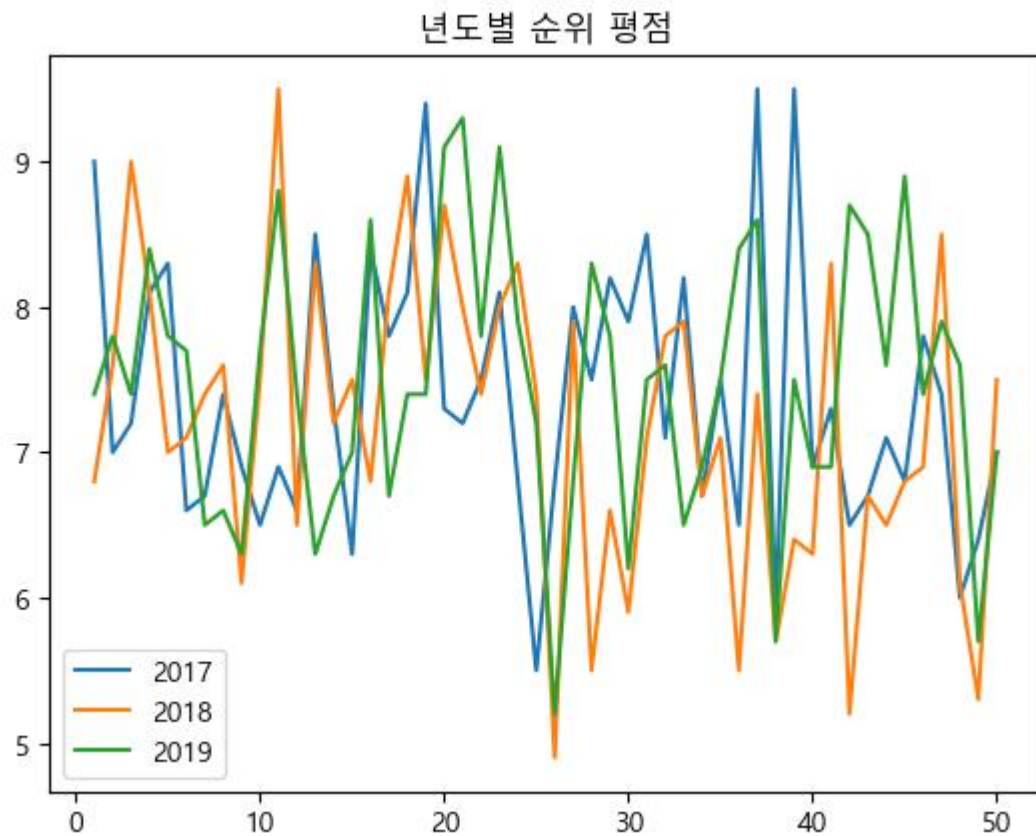
1	2	3	4
극한직업 (15)	어벤저스: 엔드게임 (12)	겨울왕국 2 (전체)	알라딘 (전체)
네티즌 ★ 7.4 19.01.23 개봉	네티즌 ★ 7.8 19.04.24 개봉	네티즌 ★ 7.4 19.11.21 개봉	네티즌 ★ 8.4 19.05.23 개봉

['극한직업',
1,
'http://t1.daumcdn.net/
movie/4e00e81f2b6f4d2
eb65b3387240cc3c0154
7608409838',
7.4,
'2019.01.23',
2019]



실습

- **년도별 평점 그래프**



실습

- 네이버 영화 사이트의 리뷰와 평점을 최신 데이터 50개를 추출하고 평점 평균과 평점 흐름을 그래프로 보이시오.
 - <https://movie.naver.com/movie/point/af/list.nhn?&page=1>
단, 한페이지에는 10개씩

평균 평점 : 6.9

