파이썬 Beautiful Soup

파이썬 Beautiful Soup

- **Beautiful Soup**
  - HTML 및 XML 문서 를 구문 분석하기위한 Python 패키지
- **Beautiful Soup 기본 사용법**
  - 패키지 읽어 들이기
  - 분석대상 지정
  - 인스턴스 생성
  - 원하는 부분 추출

```
1   from bs4 import BeautifulSoup
2   import func
3
4   html = func.fileToStr("01_html_css.html")
5
6   bs = BeautifulSoup(html, "html.parser")
7
8   #메타 태그 가져오기
9   meta = bs.meta
10  print(meta)
11  print(type(meta))
```

```
<meta charset="utf-8"/>
<class 'bs4.element.Tag'>
```

Beautiful Soup 태그 파싱

- .find(태그명)
  - 조건에 맞는 태그 1개만 찾음
- .find_all(태그명)
  - 조건에 맞는 모드 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 다
<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
--------------------
```

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 복제하고 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/">
<img alt="자유소프트웨어재단" src="https://static.fsf.org/common/img/logo-new.png"/>
</a>
--------------------
원칙1
원칙2
원칙3
--------------------
```

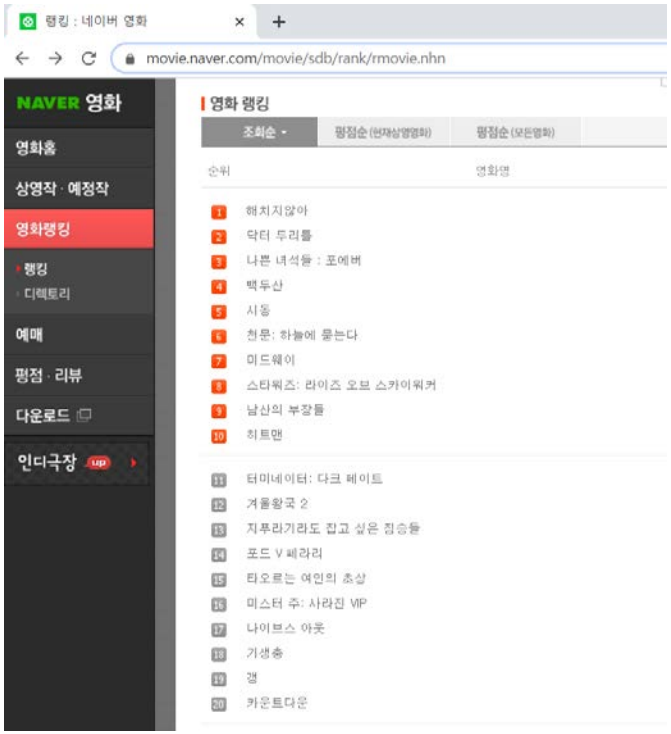
실습

- 그림의 위치를 추출하시오.

https://static.fsf.org/common/img/logo-new.png

실습

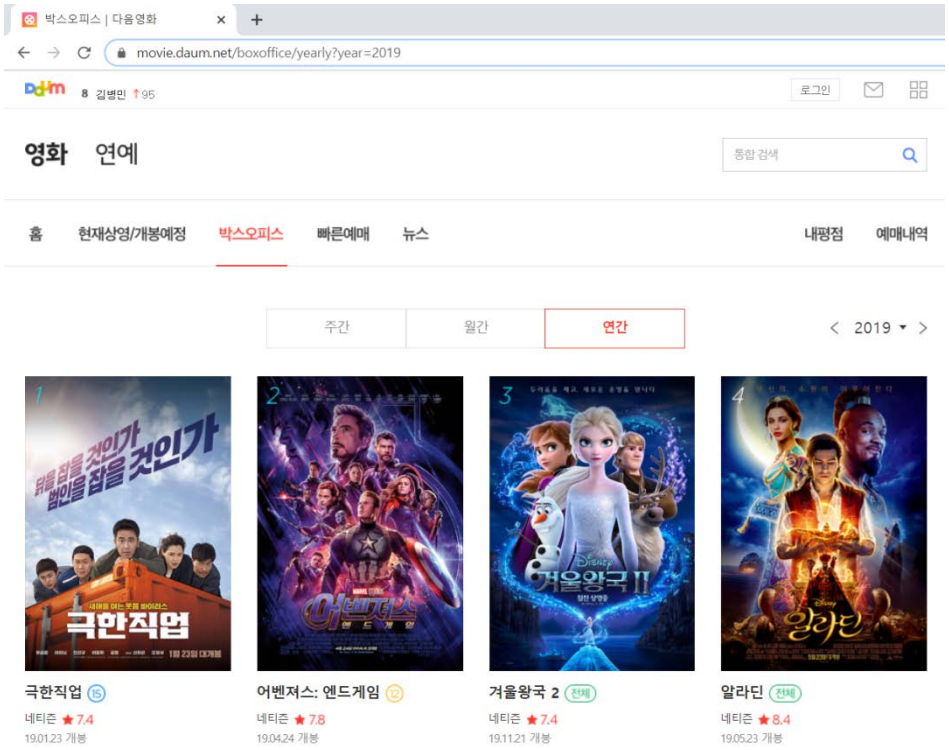
- 네이버 영화 사이트에서 영화 순위를 화면에 표시하시오.




```
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: 나이브스 아웃
18 위: 기생충
19 위: 갱
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.

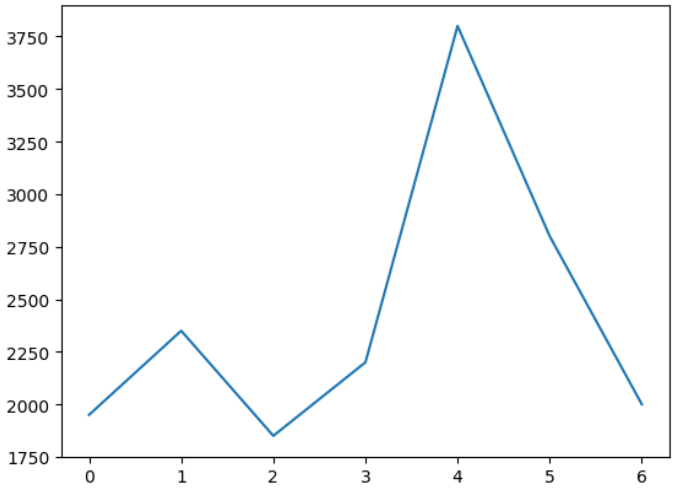


['극한직업',
1,
'http://t1.daumcdn.net/movie/4e00e81f2b6f4d2eb65b3387240cc3c01547608409838',
7.4,
'2019.01.23',
2019]

데이터 시각화

- 시각화 도구
  - Matplotlib.pyplot

```
import matplotlib.pyplot as plt

y = [1950,2350,1850, 2200,3800,2800,2000]

plt.plot(y)
plt.show()
```



데이터 시각화

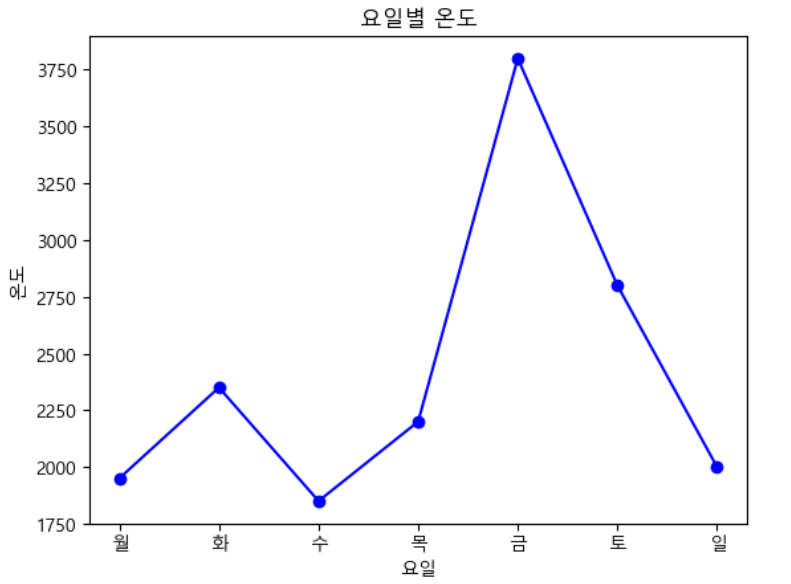
- **X축 표기 및 한글화**

```
import matplotlib.pyplot as plt

#한글 폰트 사용
from matplotlib import font_manager, rc
font_name =
font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)

x = [0,1,2,3,4,5,6]
x2 = ['월','화','수','목','금','토','일']
y = [1950,2350,1850, 2200,3800,2800,2000]

plt.plot(y)
plt.xticks(x, x2)
plt.show()
```

데이터시각화

- ## 옵션 및 레이블

```
import matplotlib.pyplot as plt

#한글 폰트 사용
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)

x = [0,1,2,3,4,5,6]
x2 = ['월','화','수','목','금','토','일']
y = [1950,2350,1850, 2200,3800,2800,2000]

plt.plot(y, 'bo-')
plt.xticks(x, x2)

plt.xlabel('요일')
plt.ylabel('온도')
plt.title('요일별 온도')
plt.show()
```



데이터시각화

- **속성명**

스타일 문자열	약자	의미
color	c	선 색깔
linewidth	lw	선 굵기
linestyle	ls	선 스타일
marker		마커 종류
markersize	ms	마커 크기
markeredgecolor	mec	마커 선 색깔
markeredgewidth	mew	마커 선 굵기
markerfacecolor	mfc	마커 내부 색깔

데이터시각화-옵션 문자

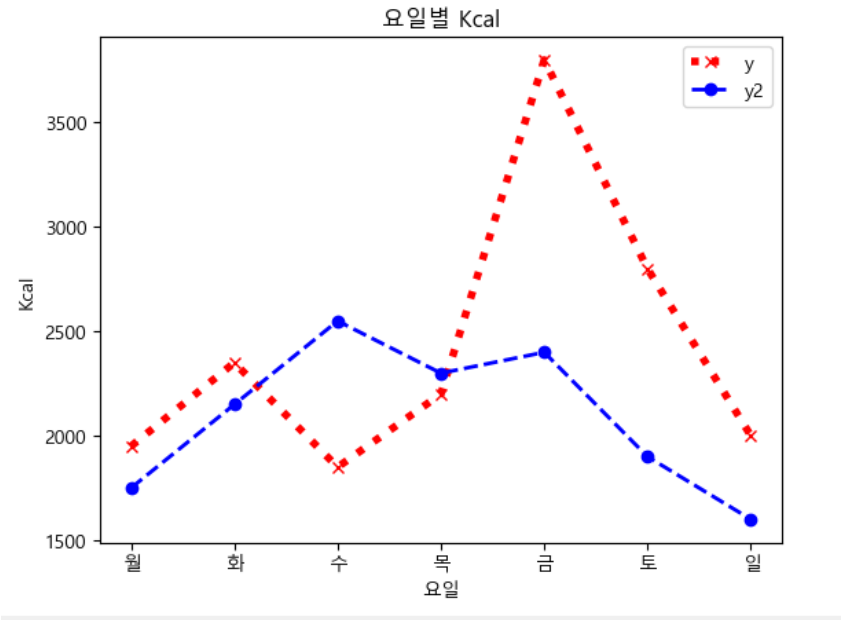
색상 문자열	약자
blue	b
green	g
red	r
cyan	c
magenta	m
yellow	y
black	k
white	w

선 스타일 문자열	의미
-	solid line style
--	dashed line style
-.	dash-dot line style
:	dotted line style

마커 문자열	의미
.	point marker
,	pixel marker
o	circle marker
v	triangle_down marker
^	triangle_up marker
<	triangle_left marker
>	triangle_right marker
1	tri_down marker
2	tri_up marker
3	tri_left marker
4	tri_right marker
s	square marker
p	pentagon marker
*	star marker
h	hexagon1 marker
H	hexagon2 marker
+	plus marker
x	x marker
D	diamond marker
d	thin_diamond marker

데이터 시각화

```
import matplotlib.pyplot as plt

#한글 폰트 사용
from matplotlib import font_manager, rc
font_name =
font_manager.FontProperties(fname="c:/Windows/Fonts/malgu
n.ttf").get_name()
rc('font', family=font_name)

x = [0,1,2,3,4,5,6]
x2 = ['월','화','수','목','금','토','일']
y = [1950,2350,1850, 2200,3800,2800,2000]
y2 = [1750,2150,2550, 2300,2400,1900,1600]

plt.plot(x, y, label='y', c='r', lw=4, ls=':', marker='x')
plt.plot(x, y2, label='y2', c='b', lw=2, ls='--', marker='o')
plt.xticks(x, x2)

plt.xlabel('요일')
plt.ylabel('온도')
plt.title('요일별 온도')
plt.show()
```

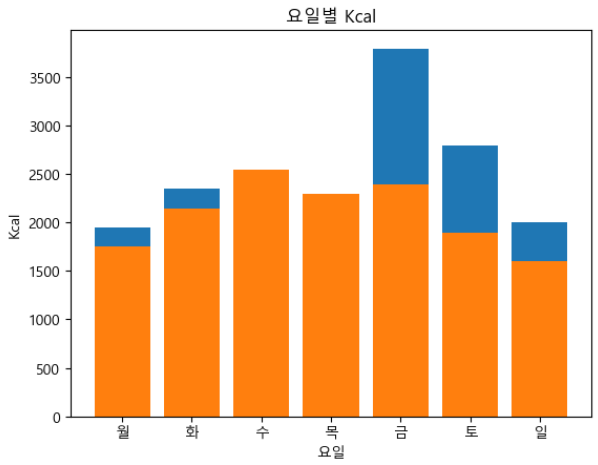


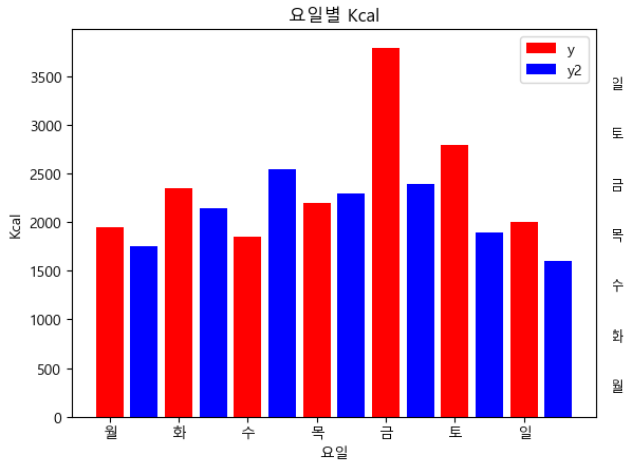
데이터시각화

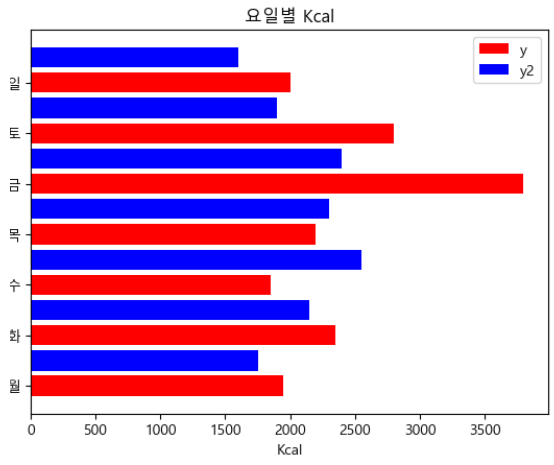
```
x = [0,1,2,3,4,5,6]
x2 = ['월','화','수','목','금','토','일']
y = [1950,2350,1850, 2200,3800,2800,2000]
y2 = [1750,2150,2550, 2300,2400,1900,1600]

plt.bar(x, y)
plt.bar(x, y2)
plt.xticks(x, x2)
```

```
x = [0,2,4,6,8,10,12]
x1 = [1,3,5,7,9,11,13]
x2 = ['월','화','수','목','금','토','일']
y = [1950,2350,1850, 2200,3800,2800,2000]
y2 = [1750,2150,2550, 2300,2400,1900,1600]

plt.bar(x, y, label='y', color='r')
plt.bar(x1, y2, label='y2', color='b')
plt.xticks(x, x2)
```

```
plt.barh(x, y, label='y', color='r')
plt.barh(x1, y2, label='y2', color='b')
plt.yticks(x, x2)

plt.ylabel('요일')
plt.xlabel('Kcal')
plt.title('요일별 Kcal')
plt.legend()
plt.show()
```

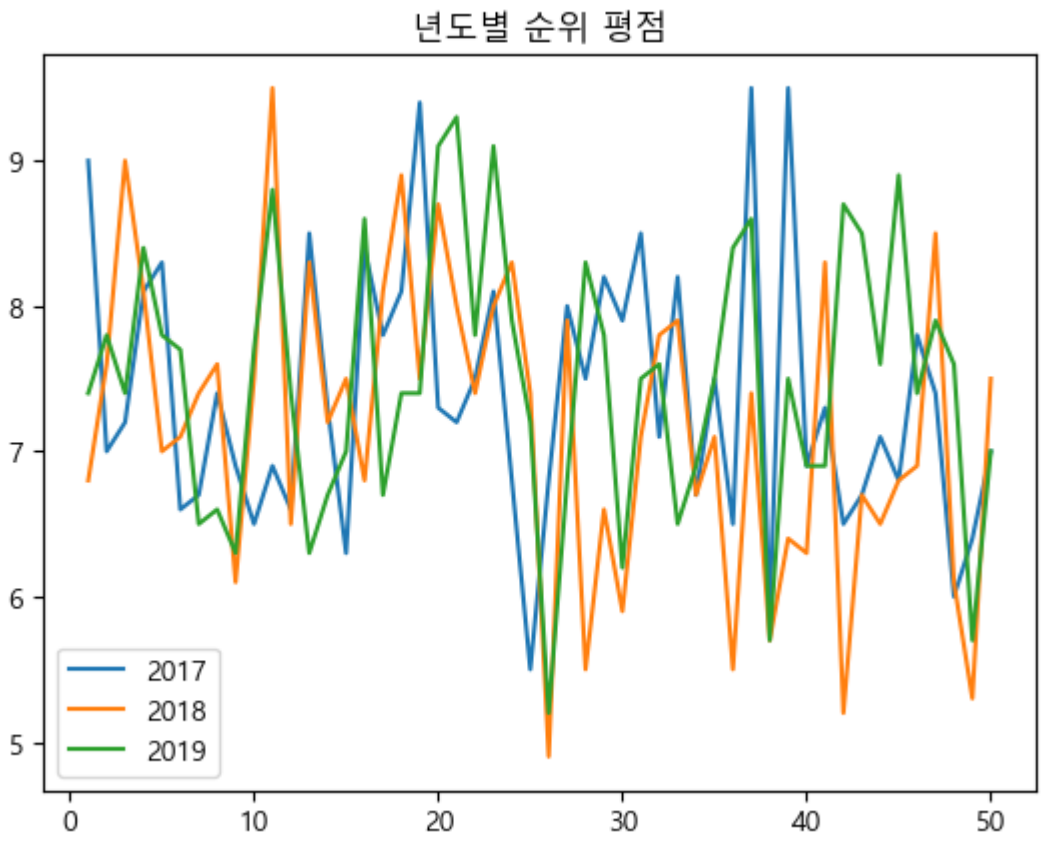






실습

- 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