파이썬 라이브러리를 활용한 데이터 분석

툴 스터디 4주차

앞으로의 커리큘럼 소개!

- 다음 툴 스터디 시간부터는 **크롤링**과 **시각화**를 배우게 됩니다.

- 우리 프로젝트에서 크롤링과 시각화 작업을 진행하지 않더라도,
  크롤링과 시각화는 재미있는 데이터 분석 과정이고, 유용하게 활용할 수 있기 때문에,
  툴 스터디 시간을 이용해서 여러분들이 크롤링/시각화 기초를 익힐 수 있도록 도움을 드리고 싶었어요!

- 오늘은 우리가 본격적으로 **뷰티풀숲 (Beautiful Soup), 셀레니움 (Selenium), 워드 클라우드 (Word Cloud)**를 다뤄보기 전에, 준비해야할 사항을 안내해드리겠습니다.
- 앞으로의 스터디에 필요한 강의 자료와 **Visual Studio Code** 프로그램 설치를 안내해 드리겠습니다.

Beautiful Soup

- 웹 데이터를 분석할 수 있는 라이브러리입니다.

```
from bs4 import BeautifulSoup
```

- 웹 페이지의 구조도인 html 파일과 xml 파일을 분석하고,
- 크롤링하고자 하는 웹 페이지의 url을 열어 웹 데이터를 분석할 수 있습니다.

Beautiful Soup

- html 파일

```
<html>
 <head>
  <title>
   The Dormouse's story
  </title>
 </head>
 <body>
  <p class="title">
   <b>
    The Dormouse's story
   </b>
  </p>
  <p class="story">
   Once upon a time there were three little sisters; and their names were
   <a class="sister" href="http://example.com/elsie" id="link1">
    Elsie
   </a>
   ,
   <a class="sister" href="http://example.com/lacie" id="link2">
    Lacie
   </a>
   and
   <a class="tillie" href="http://example.com/elsie" id="link3">
    Tillie
   </a>
   ;
and they lived at the bottom of a well.
  </p>
  <p class="story">
   ...
  </p>
 </body>
</html>
```

1) 내용 추출

```
soup.title
```

```
<title>The Dormouse's story</title>
```

```
soup.title.string
```

```
"The Dormouse's story"
```

```
soup.title.get_text()
```

```
"The Dormouse's story"
```

2) 내용 검색

```
soup.find_all('a')
```

```
[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,
 <a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,
 <a class="tillie" href="http://example.com/elsie" id="link3">Tillie</a>]
```

```
soup.find_all('a')[0].get_text()
```

```
'Elsie'
```

```
for link in soup.find_all('a'):
    print(link.get('href'))
```

```
http://example.com/elsie
http://example.com/lacie
http://example.com/elsie
```

Beautiful Soup

- xml file

```
from urllib import request
from bs4 import BeautifulSoup

target = request.urlopen("http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108")

soup = BeautifulSoup(target, "html.parser")
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<rss version="2.0">
  ▼<channel>
    <title>기상청 육상 중기예보</title>
    <link>http://www.kma.go.kr/weather/forecast/mid-term_01.jsp</link>
    <description>기상청 날씨 웹서비스</description>
    <language>ko</language>
    <generator>기상청</generator>
    <pubDate>2021년 07월 28일 (수)요일 06:00</pubDate>
    ▼<item>
      <author>기상청</author>
      <category>육상중기예보</category>
      <title>전국 육상 중기예보 - 2021년 07월 28일 (수)요일 06:00 발표</title>
      <link>http://www.kma.go.kr/weather/forecast/mid-term_01.jsp</link>
      <guid>http://www.kma.go.kr/weather/forecast/mid-term_01.jsp</guid>
      ▼<description>
        ▼<header>
          <title>전국 육상중기예보</title>
          <tm>202107280600</tm>
          ▼<wf>
            <![CDATA[ ○ (강수) 31일(토)은 전국(경남권과 강원영동 제외)에, 8월 2일(월)~3일(화)은 강원영동에 비가 오겠습니다.<br /> 한편, 8월 1일(일)과 2일(월)은
            낮 기온은 30~35도로 며제(27일, 아침최저기온 23~28도, 낮최고기온 31~36도)와 비슷하거나 조금 낮겠습니다.<br />○ (주말전망) 31일(토)은 전국(경남권과 강
            있겠습니다.<br /> 아침 기온은 24~26도, 낮 기온은 31~35도가 되겠습니다.<br />* 당분간 전국 대부분 지역에서 낮 기온이 35도 내외로 폭염이 지속되고 열대[
            의 이동경로와 강도변화에 따라 강수의 변동성이 크겠으니, 앞으로 발표되는 기상정보를 참고하기 바랍니다. ]]>
          </wf>
        </header>
        ▼<body>
          ▼<location wl_ver="3">
            <province>서울,인천,경기도</province>
            <city>서울</city>
            ▼<data>
              <mode>A02</mode>
              <tmEf>2021-07-31 00:00</tmEf>
              <wf>맑음</wf>
              <tmn>25</tmn>
              <tmx>34</tmx>
              <reliability/>
              <rnSt>20</rnSt>
            </data>
            ▼<data>
              <mode>A02</mode>
```


```
for location in soup.select("location"):
    print("도시:", location.select_one("city").string)
    print("날씨:", location.select_one("wf").string)
    print("최저기온:", location.select_one("tmn").string)
    print("최고기온:", location.select_one("tmx").string)
    print()
```

```
도시: 서울
날씨: 구름많음
최저기온: -3
최고기온: 8

도시: 인천
날씨: 구름많음
최저기온: -3
최고기온: 5

도시: 수원
날씨: 구름많음
```

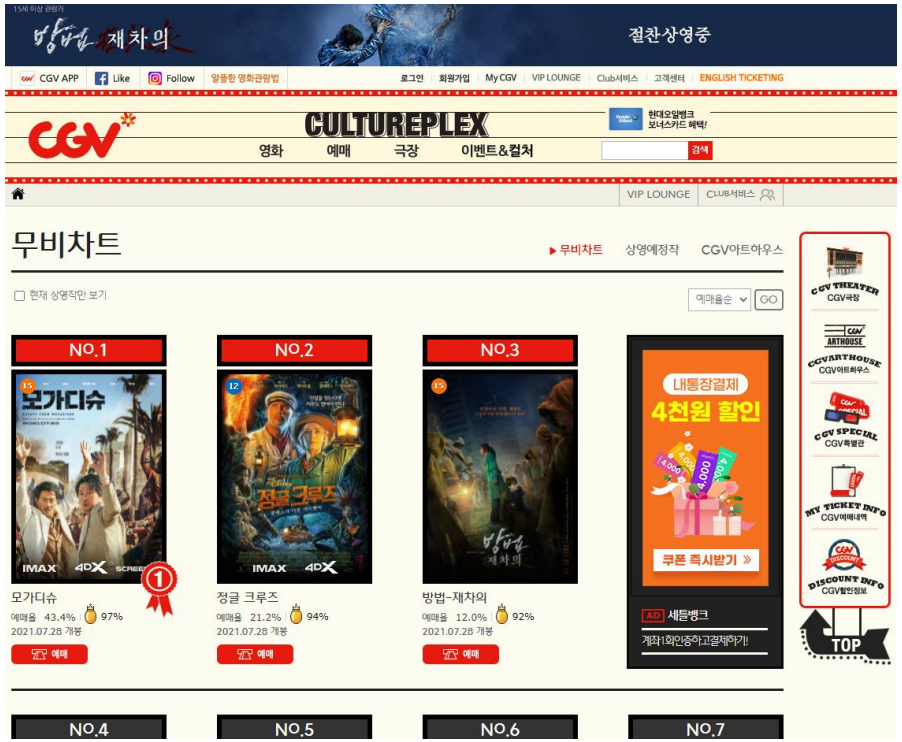
Beautiful Soup

- url Open

```
import requests
from bs4 import BeautifulSoup

r = requests.get("http://www.cgv.co.kr/movies/?ft=0")
c = r.content

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




```
ol = html.find("ol")
li = ol.find_all("li")

for l in li:
    div = l.find("div", {"class" : "box-contents"})
    strong = div.find("strong").text
    print(strong)
```

```
모가디슈
정글 크루즈
방법-재차의
```

Selenium

- 데이터 분석은 우리가 입증하고자 하는 가설을 세워 데이터 분석을 통해 가설이 합당한지, 그렇지 않은지 확인하기 위해 진행합니다.

- 오늘은 기초적인 확률·통계 개념을 배운 다음,

  개념을 바탕으로 데이터 분석 과정의 핵심인 '추론'과 '가설 검정' 과정을 이해해보겠습니다!

Word Cloud

- 키워드를 시각화하는 라이브러리

```
from konlpy.tag import Hannanum
from collections import Counter
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

- txt 텍스트 파일을 분석하여 수많은 텍스트 중 중요한 키워드들을 한눈에 파악할 수 있게 해줍니다.

Word Cloud

- txt 파일

```
text = open('2018_president_message.txt', 'r').read()

engin = Hannanum()
nouns = engin.nouns(text) # 명사
nouns = [n for n in nouns if len(n) > 1]
count = Counter(nouns)
tags = count.most_common(50)
```

존경하는 국민 여러분,

지난 일 년, 저는 평범함이 가장 위대하다는 것을
하루하루 느꼈습니다.
촛불광장에서 저는 군중이 아닌
한 사람 한 사람의 평범한 국민을 보았습니다.
어머니에서 아들로, 아버지에서 딸로 이어지는 역사가
그 어떤 거대한 역사의 흐름보다 중요하다는 것을 깨달았습니다.
한겨울 내내 촛불을 든 후 다시 일상을 충실히 살아가는
평범한 가족들을 보면서
저는 우리의 미래를 낙관할 수 있습니다.

우리가 민주주의의 역사를 다시 쓸 수 있었던 것은
그렇게 평범한 사람, 평범한 가족의 용기있는 삶이
우리 주변에 항상 존재하고 있었기 때문입니다.
저는 그것이 너무나 자랑스럽습니다.
덕분에 우리는 오늘 희망을 다시 이야기할 수 있게 되었습니다.

국민들께서는 자신의 소중한 일상을 국가에 내어주었습니다.
나라를 바로 세울 힘을 주었습니다.
이제 국가는 국민들에게 응답해야 합니다.
더 정의롭고, 더 평화롭고, 더 안전하고, 더 행복한 삶을 약속해야 합니다.
그것이 바로 나라다운 나라입니다.

2018년 새해, 정부와 저의 목표는
국민들의 평범한 일상을 지키고, 더 나아지게 만드는 것입니다.

국민의 뜻과 요구를 나침반으로 삼겠습니다.
국민들께서 삶의 변화를 체감할 수 있게 하겠습니다.

국민 여러분,
제가 대통령이 되어 제일 먼저 한 일은
직무실에 일자리 상황판을 설치한 것입니다.

```
wordcloud = WordCloud(font_path='c:/Windows/Fonts/malgun.ttf',
                      background_color='white',
                      width=1200, height=800).generate_from_frequencies(dict(tags))
fig = plt.figure(figsize=(12,12))
plt.axis('off')
plt.imshow(wordcloud)
plt.show()
```

Word Cloud

- txt 파일

```
text = open('2018_president_message.txt', 'r').read()

engin = Hannanum()
nouns = engin.nouns(text) # 명사
nouns = [n for n in nouns if len(n) > 1]
count = Counter(nouns)
tags = count.most_common(50)
```

존경하는 국민 여러분,

지난 일 년, 저는 평범함이 가장 위대하다는 것을
하루하루 느꼈습니다.
촛불광장에서 저는 군중이 아닌
한 사람 한 사람의 평범한 국민을 보았습니다.
어머니에서 아들로, 아버지에서 딸로 이어지는 역사가
그 어떤 거대한 역사의 흐름보다 중요하다는 것을 깨달았습니다.
한겨울 내내 촛불을 든 후 다시 일상을 충실히 살아가는
평범한 가족들을 보면서
저는 우리의 미래를 낙관할 수 있습니다.

우리가 민주주의의 역사를 다시 쓸 수 있었던 것은
그렇게 평범한 사람, 평범한 가족의 용기있는 삶이
우리 주변에 항상 존재하고 있었기 때문입니다.
저는 그것이 너무나 자랑스럽습니다.
덕분에 우리는 오늘 희망을 다시 이야기할 수 있게 되었습니다.

국민들께서는 자신의 소중한 일상을 국가에 내어주었습니다.
나라를 바로 세울 힘을 주었습니다.
이제 국가는 국민들에게 응답해야 합니다.
더 정의롭고, 더 평화롭고, 더 안전하고, 더 행복한 삶을 약속해야 합니다.
그것이 바로 나라다운 나라입니다.

2018년 새해, 정부와 저의 목표는
국민들의 평범한 일상을 지키고, 더 나아지게 만드는 것입니다.

국민의 뜻과 요구를 나침반으로 삼겠습니다.
국민들께서 삶의 변화를 체감할 수 있게 하겠습니다.

국민 여러분,
제가 대통령이 되어 제일 먼저 한 일은
직무실에 일자리 상황판을 설치한 것입니다.

```
wordcloud = WordCloud(font_path='c:/Windows/Fonts/malgun.ttf',
                      background_color='white',
                      width=1200, height=800).generate_from_frequencies(dict(tags))
fig = plt.figure(figsize=(12,12))
plt.axis('off')
plt.imshow(wordcloud)
plt.show()
```

Word Cloud

- 이미지를 이용해 원하는 형태로 시각화할 수 있습니다.

```
from konlpy.tag import Hannanum
from collections import Counter
import matplotlib.pyplot as plt
from wordcloud import WordCloud, ImageColorGenerator
from PIL import Image
import numpy as np

mask = np.array(Image.open('09. heart.jpg'))
image_colors = ImageColorGenerator(mask)
```

```
wordcloud = WordCloud(font_path='c:/Windows/Fonts/malgun.ttf',
                      relative_scaling = 0.1, mask = mask,
                      background_color='white',
                      min_font_size = 1,
                      max_font_size = 100).generate_from_frequencies(dict(tags))
fig = plt.figure(figsize=(12,12))
plt.axis('off')
plt.imshow(wordcloud.recolor(color_func=image_colors), interpolation='bilinear')
plt.show()
```

다음주!

- Beautiful Soup으로 뉴스 크롤링 실습