

LESSON

matplotlib

파이썬 시각화 라이브러리

1. Matplotlib: 가장 널리 사용되는 시각화 라이브러리 중 하나입니다. 다양한 차트 및 플롯을 생성할 수 있고, 강력하면서도 유연한 기능을 제공.

- 웹사이트: [Matplotlib](#)

2. Seaborn: Matplotlib을 기반으로 한 통계 데이터 시각화 라이브러리입니다. 간단한 코드로 다양한 통계 그래픽을 그릴 수 있으며, 기본적으로 예쁜 스타일을 제공.

- 웹사이트: [Seaborn](#)

3. Plotly: 인터랙티브한 웹 기반 시각화를 제공하는 라이브러리입니다. 차트 및 그래프를 생성하고 웹 브라우저에서 인터랙티브하게 탐색할 수 있음.

- 웹사이트: [Plotly](#)

4. Bokeh: 웹 기반 시각화를 지원하는 라이브러리로, 인터랙티브 플롯 및 대시보드를 만들 수 있음.

- 웹사이트: [Bokeh](#)

5. Altair: 간단하면서도 선언적인 문법을 사용하여 인터랙티브한 시각화를 생성할 수 있는 라이브러리.

- 웹사이트: [Altair](#)

6. Pandas Plotting: Pandas에 내장된 간단한 플로팅 기능을 제공합니다. Pandas 데이터프레임에서 직접 시각화할 수 있음.

- 웹사이트: [Pandas Visualization](#)

<https://matplotlib.org>

<https://wikidocs.net/book/5011>

1. 라이브러리 설치

```
pip install matplotlib
```

2. 라이브러리 import

```
import matplotlib.pyplot as plt
```

3. 데이터로 그래프 그리기

```
plt.plot([1,2,3,4,5])
```

4. 그래프 표시하기

```
plt.show()
```

```
import matplotlib.pyplot as plt

score=[65,100,80,85,90,55,30,67,78,88]
name=['a','b','c','d','e','f','g','h','i','j']

# 1.꺾은선 그래프 plot
plt.plot(score)
plt.show()

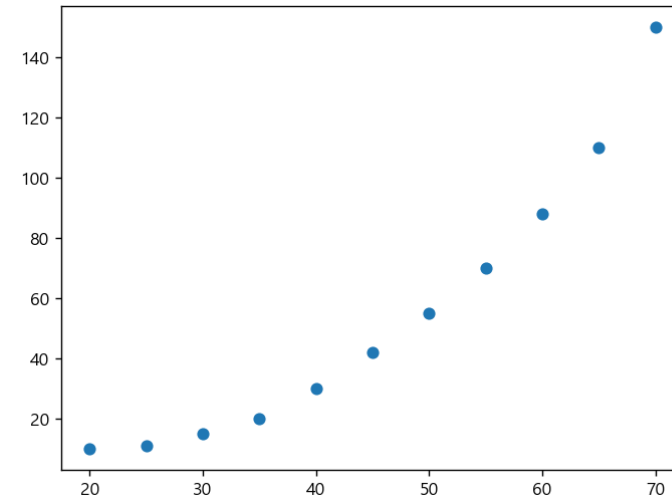
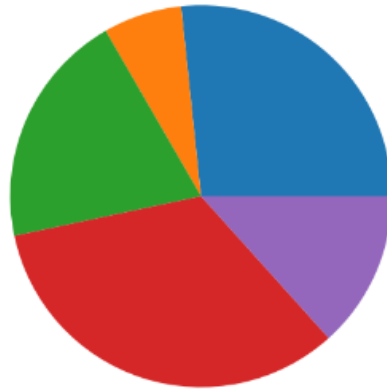
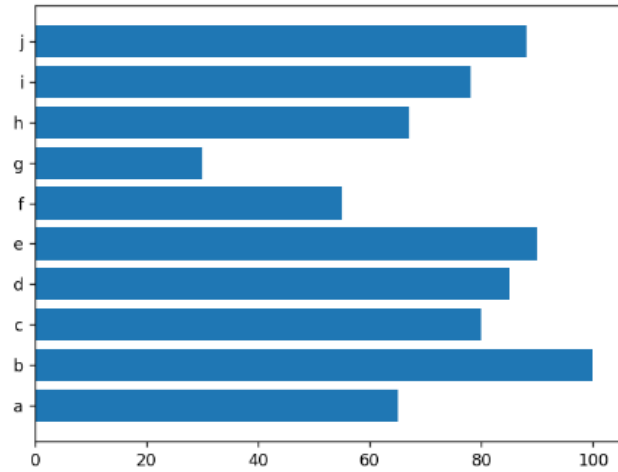
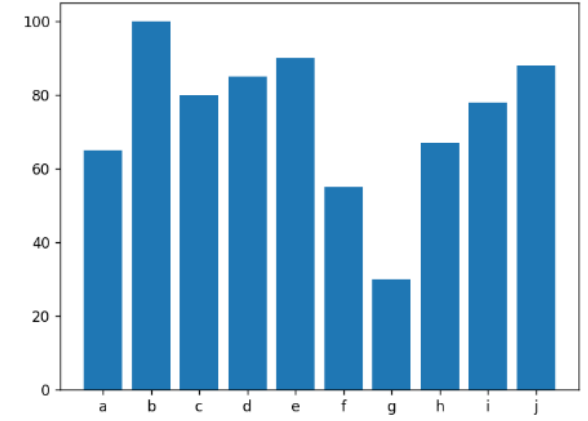
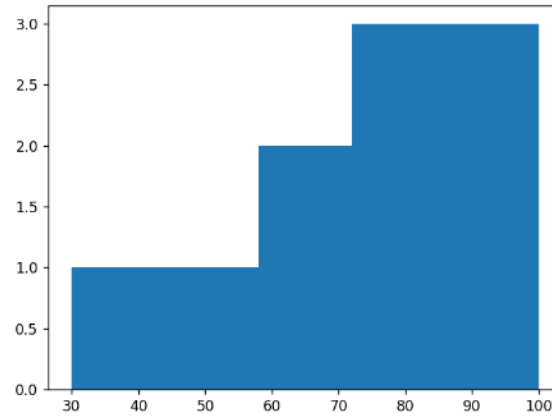
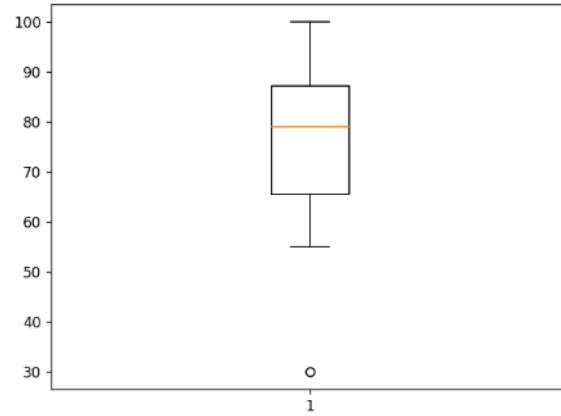
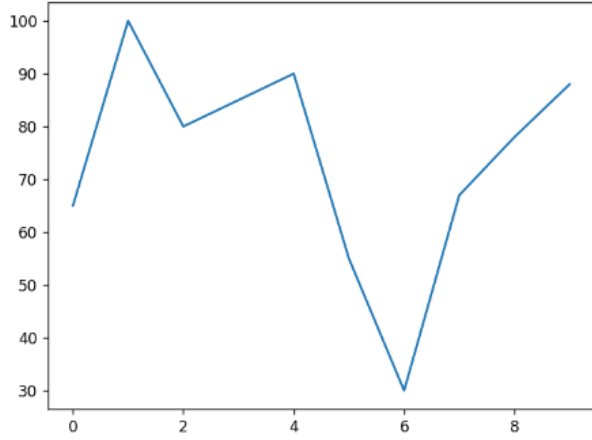
# 2.박스플롯 그래프 boxplot
plt.boxplot(score)
plt.show()

# 3.히스토그램 그래프 hist
plt.hist(score,bins=5) # bins: 계급의 구간수 지정
plt.show()

# 4.막대 그래프 bar, barh -가로축, 세로축 지정할 것
plt.bar(name,score)
plt.show()
plt.barh(name,score)
plt.show()
```

```
# 5.원형 그래프 pie
data=[4,1,3,5,2]
plt.pie(data)
plt.show()

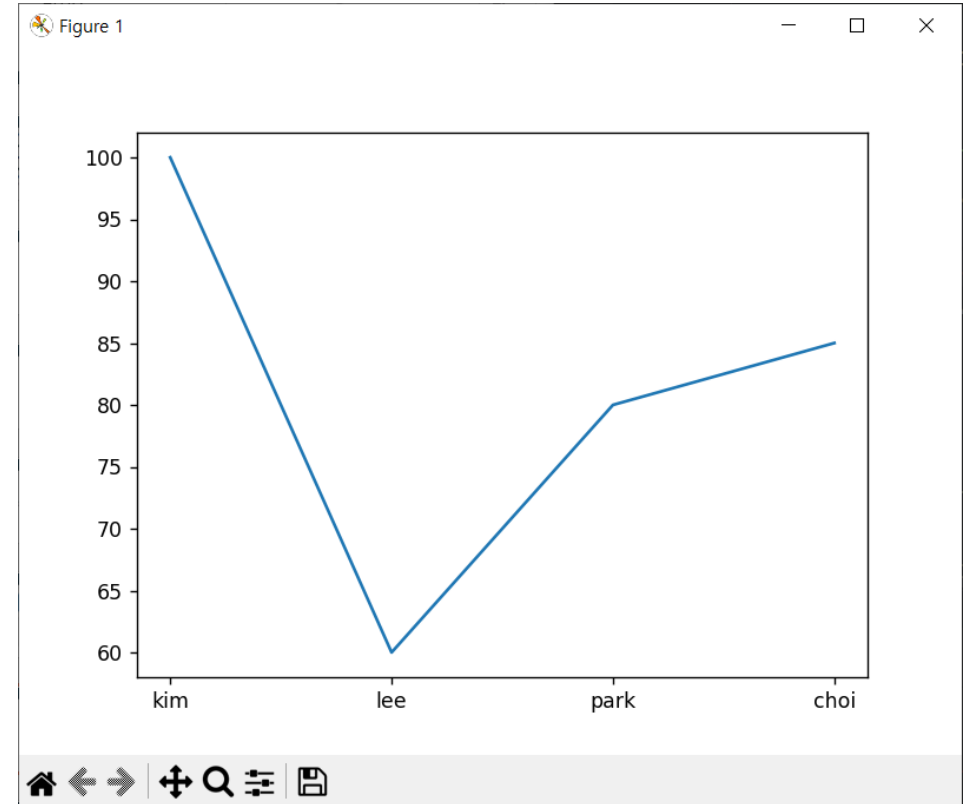
# 6.산점도 그래프 scatter
noise=[20,25,30,35,40,45,50,55,60,65,70]
stress=[10,11,15,20,30,42,55,70,88,110,150]
plt.scatter(noise,stress)
plt.show()
```



```
import matplotlib.pyplot as plt

name=['kim','lee','park','choi']
score=[100,60,80,85]

plt.plot(name,score)
# plt.pie(score)
plt.show()
```



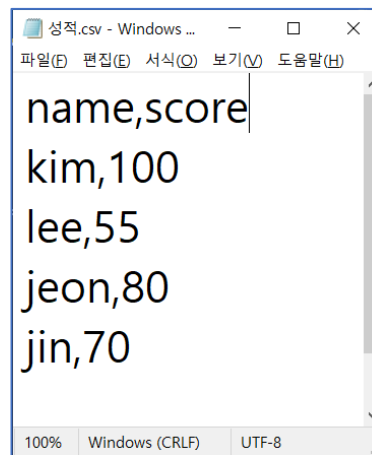
실습2(파일 데이터 시각화)

```
import matplotlib.pyplot as plt
import csv

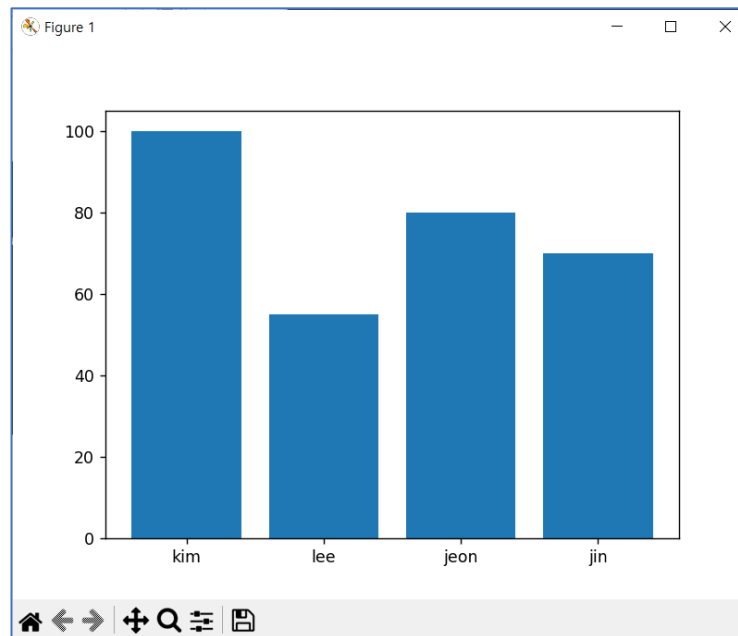
data=csv.reader(open('성적.csv',encoding='utf-8'))
next(data)
names=[]
scores=[]
for row in data:
    names.append(row[0])
    scores.append(int(row[1]))
```

```
plt.bar(names,scores)
plt.show()
```

```
plt.bar(names,scores)
plt.savefig("성적현황.png") # 이미지로 저장
plt.show()
```



| name | score |
|------|-------|
| kim | 100 |
| lee | 55 |
| jeon | 80 |
| jin | 70 |



구글 코랩에서 차트 한글 깨짐 해결하기

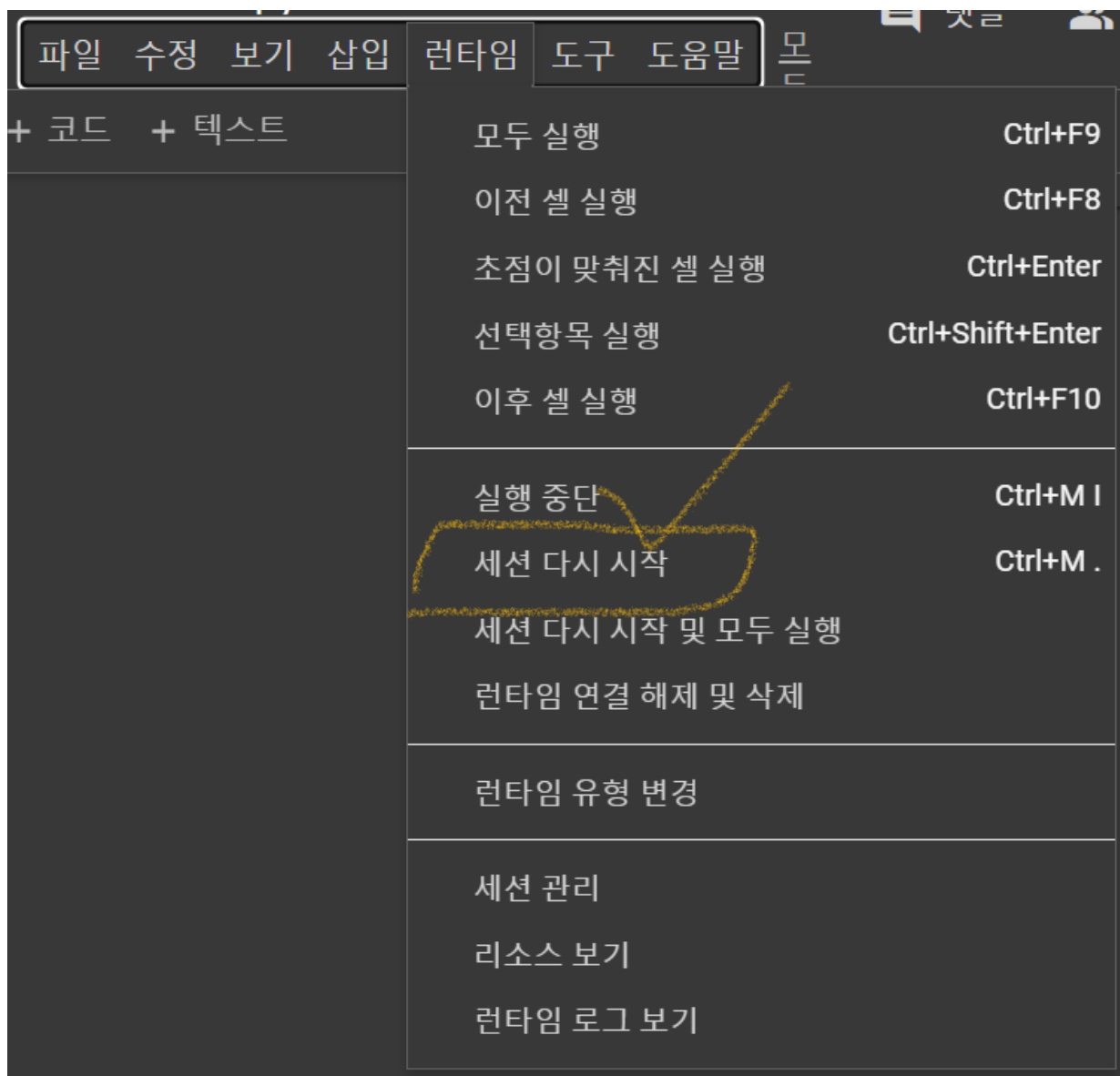
1. 폰트 설치(나눔 폰트 설치)

```
<python />  
  
!sudo apt-get install -y fonts-nanum  
!sudo fc-cache -fv  
!rm ~/.cache/matplotlib -rf
```

2. 폰트 설정

```
import matplotlib.pyplot as plt
plt.rc('font', family='NanumBarunGothic')
plt.rcParams['axes.unicode_minus'] = False
```

3. 런타임 재시작(런타임-> 세션 다시 시작)



한글과 마이너스 기호가 있는 경우 해결법

실습(시스템 글꼴 이름 확인하기)

1. c:\windows\fonts 폴더에서 "굴림 보통"을 선택하고 마우스 우클릭
2. "속성"을 선택하면 글꼴 파일이름이 표시된다.
단, "맑은 고딕"같은 "속성"이 없는 경우 더블 클릭하면 속성을 볼 수 있다.

```
from matplotlib import font_manager as fm
```

```
# 맑은 고딕의 글꼴 파일명
```

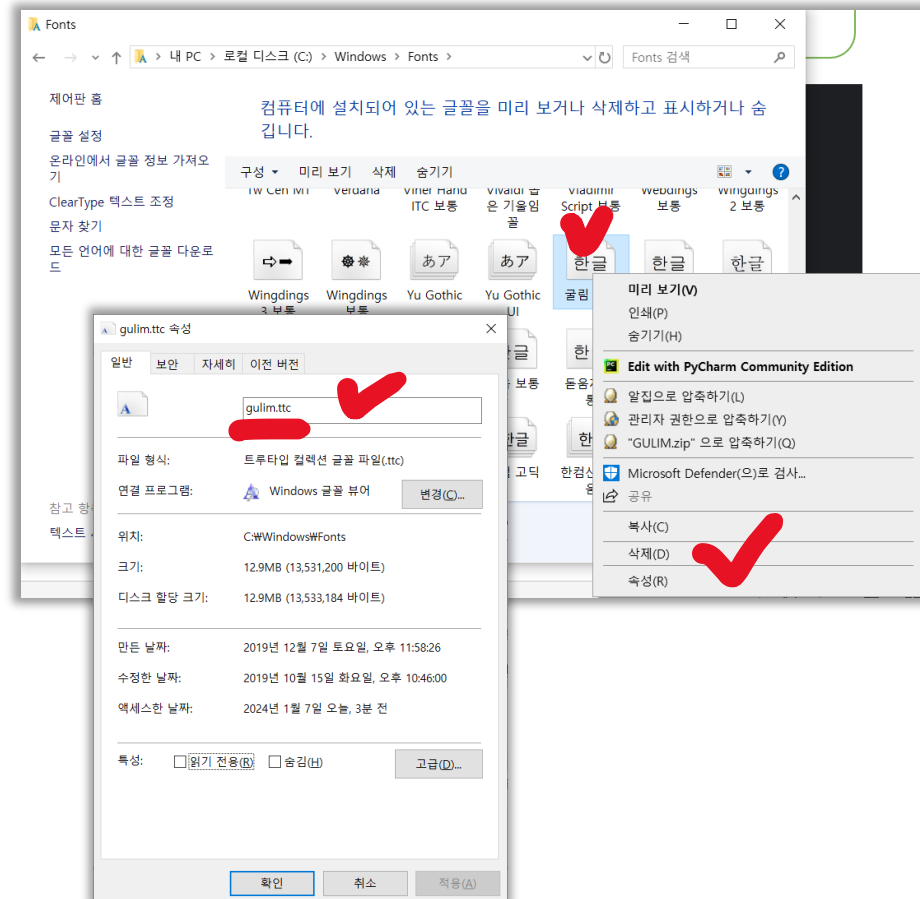
```
font_path=r"c:\windows\fonts\malguns1.ttf"
```

```
# 휴먼 편지체의 글꼴 파일명
```

```
# font_path=r"c:\windows\fonts\HMFMPYUN.TTF"
```

```
font_name=fm.FontProperties(fname=font_path).get_name()
```

```
print(font_name) # 결과: Malgun Gothic
```



- 한글 글꼴 설정 : `plt.rc('font', family='malgun gothic')`
- 마이너스 기호 설정 : `plt.rcParams['axes.unicode_minus']=False`

※ 참고

rc() 함수는 Matplotlib의 설정을 변경하는 데 사용되는 함수이다.

rc는 "run command"의 약자로, 여러 설정을 한 번에 변경하거나 특정 설정을 조회하고 수정할 때 사용된다.

`rc(group, **kwargs)`

- **group**: 설정을 적용할 그룹을 나타내는 문자열. 'font', 'axes', 'lines' 등이 있다.
- ****kwargs**: 설정을 변경할 키워드 인수들. 각 그룹에 따라 적용 가능한 설정들이 다르다.

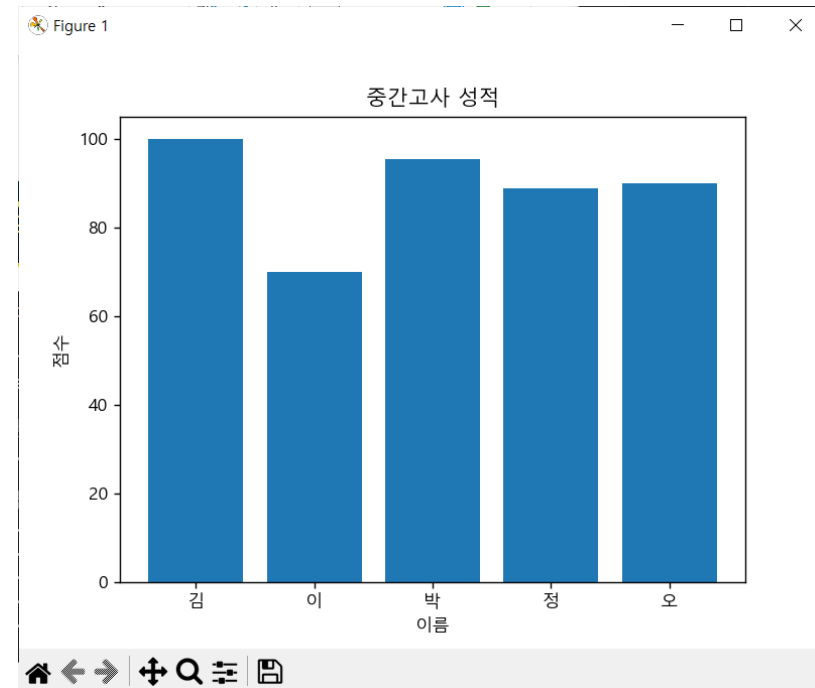
rcParams는 Matplotlib의 전역 설정을 관리하는 객체입니다. 이 객체를 사용하여 그래프의 다양한 속성을 설정할 수 있습니다. 그래픽 관련 설정, 텍스트 속성, 폰트, 색상 등을 포함하는 딕셔너리 형태로 제공됩니다.

```
import matplotlib.pyplot as plt

score=[100,70,95.4,88.9,90]
name=['김','이','박','정','오']

# 한글 깨짐 처리
plt.rc("font",family="Malgun Gothic")

# 꺾은선 그래프 만들기
plt.title("중간고사 성적")      # 차트 제목
plt.xlabel("이름")             # x축 제목
plt.ylabel("점수")             # y축 제목
plt.bar(name,score)
plt.show()
```



실습(마이너스 기호 처리)

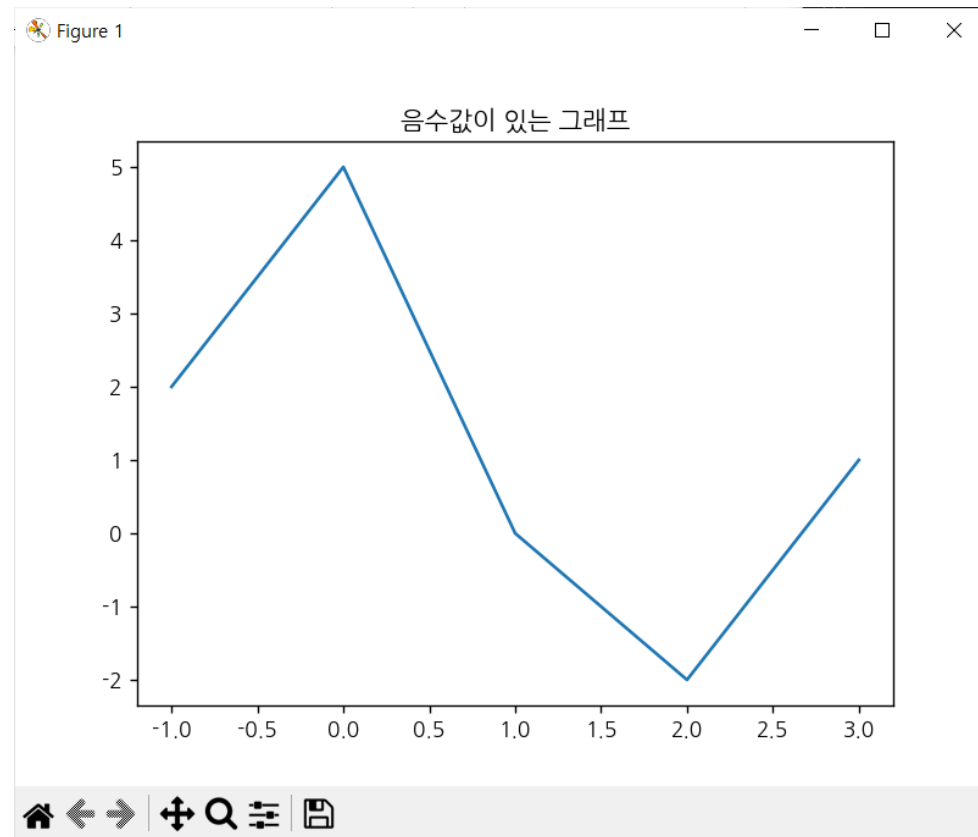
```
import matplotlib.pyplot as plt

# 한글 폰트 설정: rcParams or rc 사용
plt.rcParams['font.family'] = 'NanumGothic'
# plt.rc("font",family="Malgun Gothic")

# 음수 기호 설정
plt.rcParams['axes.unicode_minus'] = False

# 예시 데이터
x = [-1, 0, 1, 2, 3]
y = [2, 5, 0, -2, 1]

# 그래프 그리기
plt.plot(x, y)
plt.title("음수값이 있는 그래프")
plt.show()
```



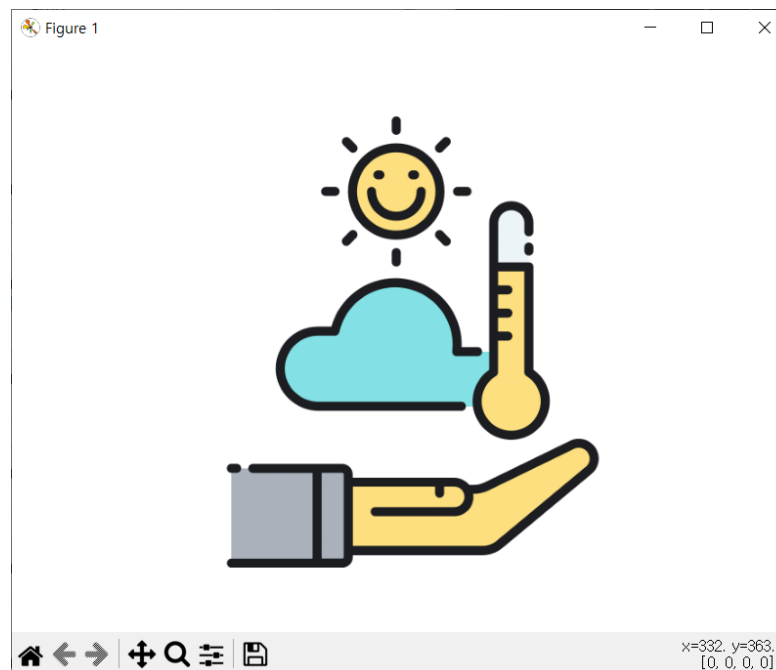
※ 한글처리를 위해 폰트 설정을 한 경우 데이터에 음수가 있는 경우 마이너스 기호가 깨지는 것을 처리할 때 `plt.rcParams['axes.unicode_minus'] = False` 설정이 필요함.

실습(차트에 이미지 표시)

```
### 이미지를 표시해보자
import matplotlib.pyplot as plt
from matplotlib.image import imread

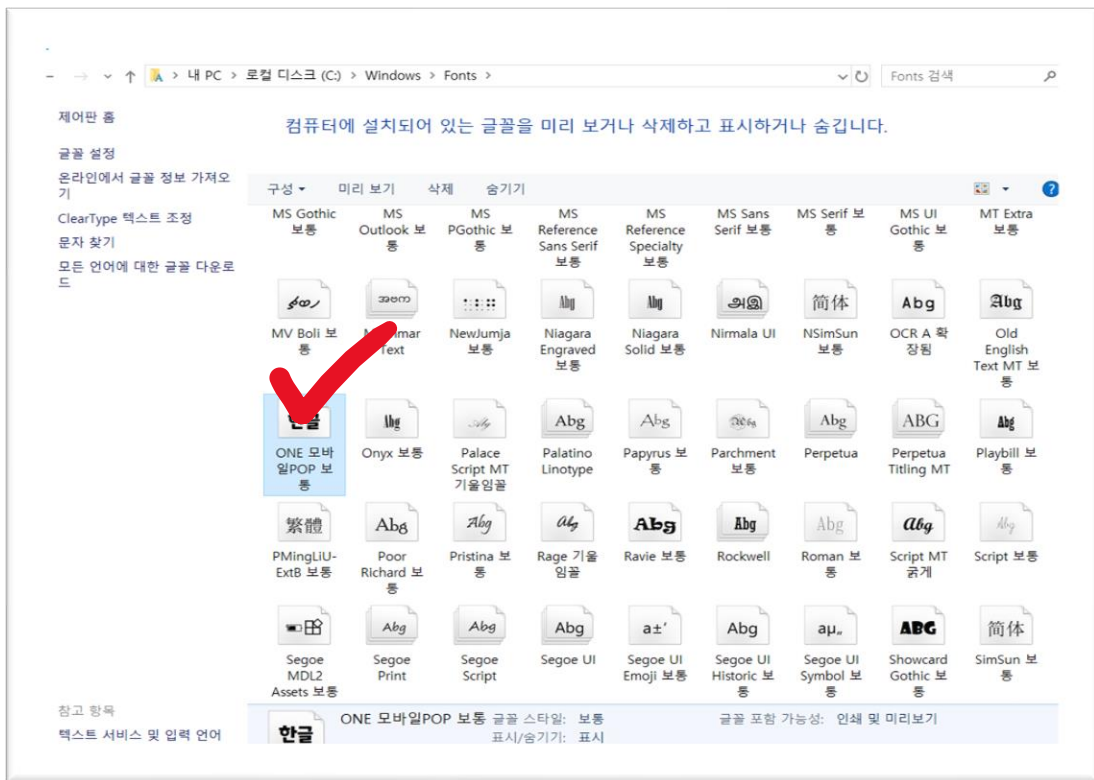
img=imread('weather.png')
plt.imshow(img)
plt.axis('off') # 축 제거
plt.show()
```

- 이미지
 - matplotlib.image 모듈의 imread() : 이미지를 읽어들이м
 - imshow() : 이미지 표시

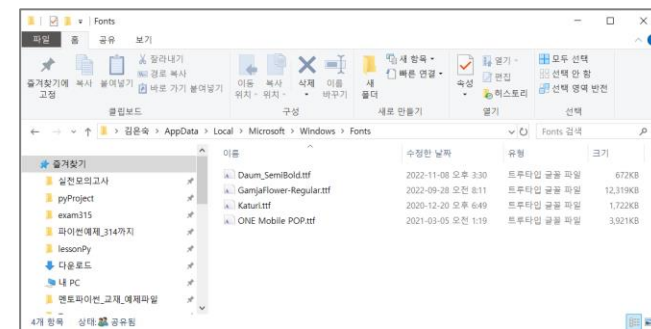
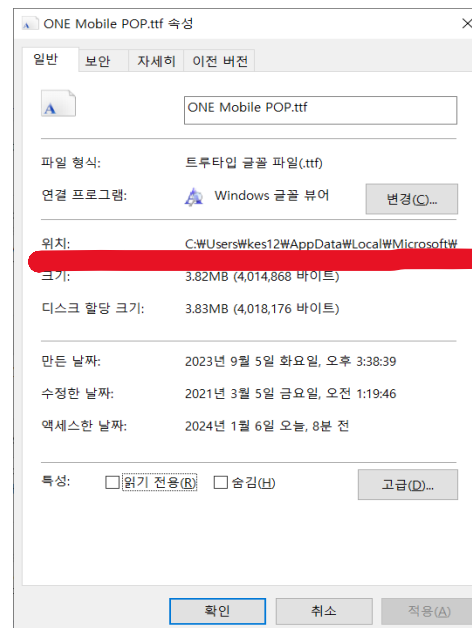


다운받은 글꼴이 설치된 경로 확인

1. c:\windows\fonts 폴더에서 원하는 글꼴 파일을 선택하고 "속성"을 클릭



2. 경로를 복사하고 경로로 이동한다.



실습(다운받아 설치한 글꼴 사용)

```
import matplotlib.pyplot as plt
from matplotlib import font_manager

font_path=r"C:\Users\kes12\AppData\Local\Microsoft\Windows\Fonts\ONE Mobile POP.ttf"
# font_path=r"c:\windows\fonts\gulim.ttc"          # 굴림의 글꼴 파일명
# font_path=r"c:\windows\fonts\HMFMPYUN.TTF"       # 휴먼 편지체의 글꼴 파일명
font_name=font_manager.FontProperties(fname=font_path).get_name()

plt.rc("font",family=font_name)
score=[100,70,95.4,88.9,90]
name=['김','이','박','정','오']

# 꺾은선 그래프 만들기
plt.title("중간고사 성적")      # 차트 제목
plt.xlabel("이름")             # x축 제목
plt.ylabel("점수")             # y축 제목
plt.bar(name,score)
plt.show()
```

※ 설치된 모든 글꼴을 사용할 수 있는 것은 아니다.

LESSON

collections 모듈의 Counter

iterable 한 객체에서 각 요소의 개수 세기

```
### collections의 Counter로 개수 세기
from collections import Counter

data=[1,2,3,4,1,1]
# data='hello, world'
c=Counter(data)
print(c)
```

```
Counter({1: 3, 2: 1, 3: 1, 4: 1})
```

```
### collections의 Counter로 개수 세기
from collections import Counter

data=[1,2,3,4,1,1,3]
# data="hello, world"
c=Counter(data)
print(c.most_common())    # 빈도수 순으로 출력
print(c.most_common(2))   # 빈도수 순으로 상위 2건 출력
```

```
[(1, 3), (3, 2), (2, 1), (4, 1)]
[(1, 3), (3, 2)]
```

LESSON

wordcloud

텍스트 데이터의 빈도수에 따라 단어를 시각적으로 표현하는 데 사용되는 라이브러리

1. 라이브러리 설치

```
pip install wordcloud
```

2. 라이브러리 import

```
import wordcloud
```

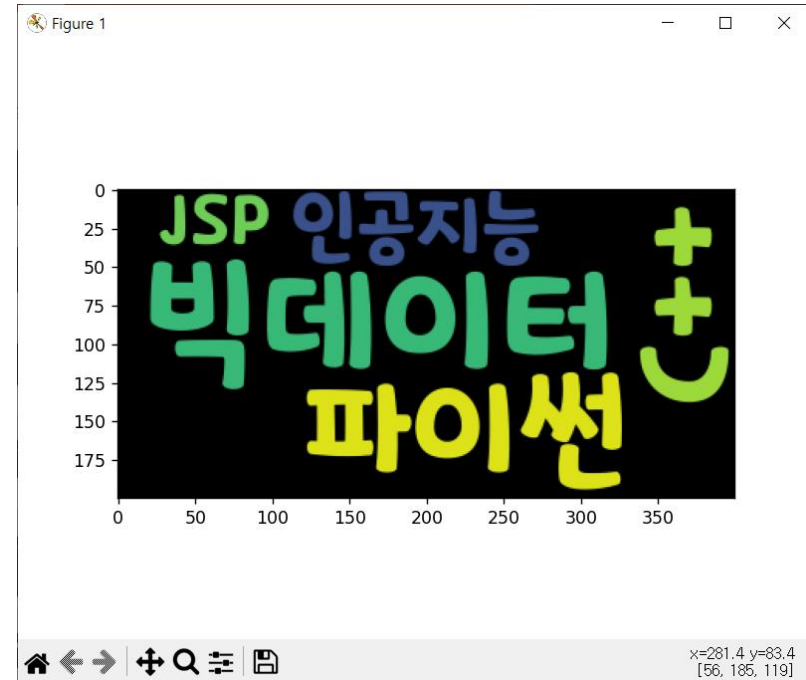
```
### wordcloud로 시각화
import wordcloud
import matplotlib.pyplot as plt

words={'파이썬':10, '빅데이터':12, '인공지능':7, 'JSP':5, 'C++':9}

# wc=wordcloud.WordCloud()
wc=wordcloud.WordCloud(font_path='Katuri.ttf') # 한글 글꼴 지정
cloud=wc.generate_from_frequencies(words)

plt.imshow(cloud)
plt.show()
```

※ Katuri.ttf 글꼴 파일이 현재 폴더에 있어야 한다.



```
### exam.txt 파일 시각화하기
from collections import Counter
import wordcloud
import matplotlib.pyplot as plt

# 사용할 폰트 파일을 미리 저장하고 아래와 같이 설정한다.
wc=wordcloud.WordCloud(font_path='Katurit.ttf')

word_list=[] # 공백으로 분리된 단어를 저장할 리스트
with open('exam.txt',encoding='utf-8') as f:
    exam=f.readlines()

for line in exam:
    words=line.strip('\n').strip('.').split(' ') # (\n)줄바꿈 문자와 "."을 제거하고 공백으로 분리
    for word in words:
        word_list.append(word)

c=Counter(word_list)
cloud=wc.generate_from_frequencies(c)
plt.imshow(cloud)
plt.show()
```

LESSON

KoNLPY로 한국어 형태소 분석

- 한국어의 자연어 처리를 위한 파이썬 라이브러리
- 형태소 분석, 명사 추출, 품사 태깅, 구문 분석 등과 같은 기능들을 제공
- 한국어 문장에서 단어를 추출하고 토큰화
- 여러 가지 형태소 분석기를 지원
제공되는 분석기 **Kkma, Hannanum, Mecab, Komoran, Okt (Open Korean Text),...**

0. 구글 코랩을 실행한다.

파이썬 한글 형태소 분석기 konlpy 내부의 클래스는 Java 기반으로 작성되어있기 때문에 반드시 JDK 가 설치되어있어야 konlpy 를 사용할 수 있다. 그런데 구글 코랩에서는 JDK설치 없이 사용 가능.

1. konlpy 라이브러리 설치하기

```
pip install konlpy
```

2. 형태소 분석기 import

```
from konlpy.tag import Okt
```

3. Okt 형태소 분석기 객체 생성

```
okt = Okt()
```

4. 문장에서 형태소와 품사 정보 얻기

```
okt.pos(문자열)
```

```
from konlpy.tag import Okt
```

```
okt=Okt()
```

```
text="파이썬은 쉬운 문법으로 초보자도 접근하기 쉬운 인터프리터 언어이다."
```

```
okt.pos(text)
```

```
# 명사만 출력하기
```

```
okt.nouns(text)
```

```
[('파이썬', 'Noun'), ('은', 'Josa'), ('쉬운', 'Adjective'), ('문법', 'Noun'), ('으로', 'Josa'), ('초보자', 'Noun'), ('도', 'Josa'), ('접근', 'Noun'), ('하기', 'Verb'), ('쉬운', 'Adjective'), ('인터프리터', 'Noun'), ('언어', 'Noun'), ('이다', 'Josa'), ('.', 'Punctuation')]
```

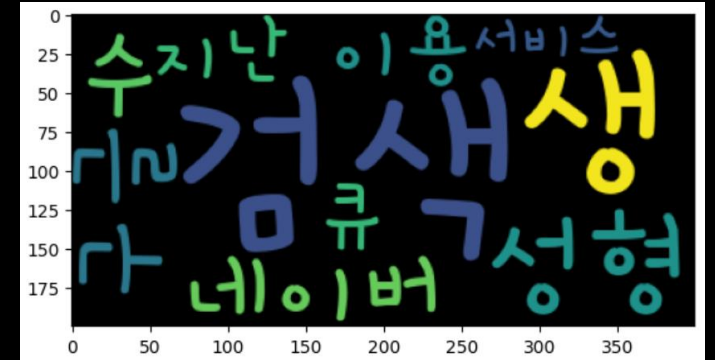
```
from konlpy.tag import Okt
from collections import Counter
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

```
okt=Okt()
```


```
text=open("n_news.txt",'r',encoding='utf8').read()
okt.pos(text)
```

```
okt_list=okt.nouns(text)      # 명사만 추출
c=Counter(okt_list)           # 명사의 단어별 개수를 리스트로 생성
tags=c.most_common(10)        # 상위 10개 단어만 추출
tags
```

```
wc=wordcloud.WordCloud(font_path='gf.ttf') # 한글 처리
wc=wc.generate_from_frequencies(dict(tags)) # 딕셔너리로 변경하기
plt.imshow(wc)
plt.show()
```



뉴스 파일에서 한글 형태소 시각화

Python 2 

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

text=open("n_news.txt",'r',encoding='utf8').read()
okt=Okt()
line=[]
line=okt.pos(text)
line
n_adj=[]
for word, tag in line:
    if tag=="Noun" and len(word)>=2:    # 명사이고 길이가 2자 이상인 것만
        n_adj.append(word)

counts=Counter(n_adj)
tags=counts.most_common(10)
wc=WordCloud(font_path='gf.ttf')
wc=wc.generate_from_frequencies(dict(tags))

plt.imshow(wc)
plt.axis('off')
plt.show()
```

