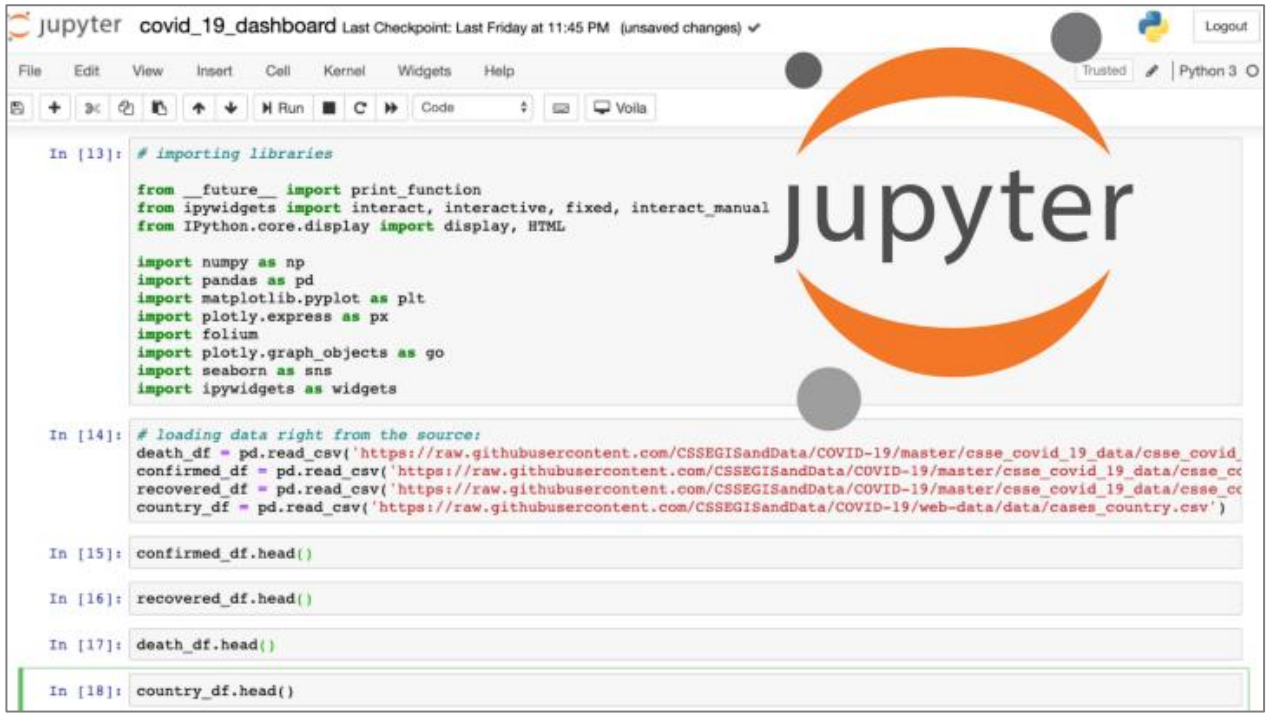
11주차
파이썬 기초 프로그래밍

Anaconda 설치

- 패키지 관리 시스템
  - 파이썬으로 개발을 할 때 필요한 다양한 패키지들을 손쉽게 관리할 있도록 도와줌
  - https://www.anaconda.com/
- 설치 및 실행
  - https://ndb796.tistory.com/355

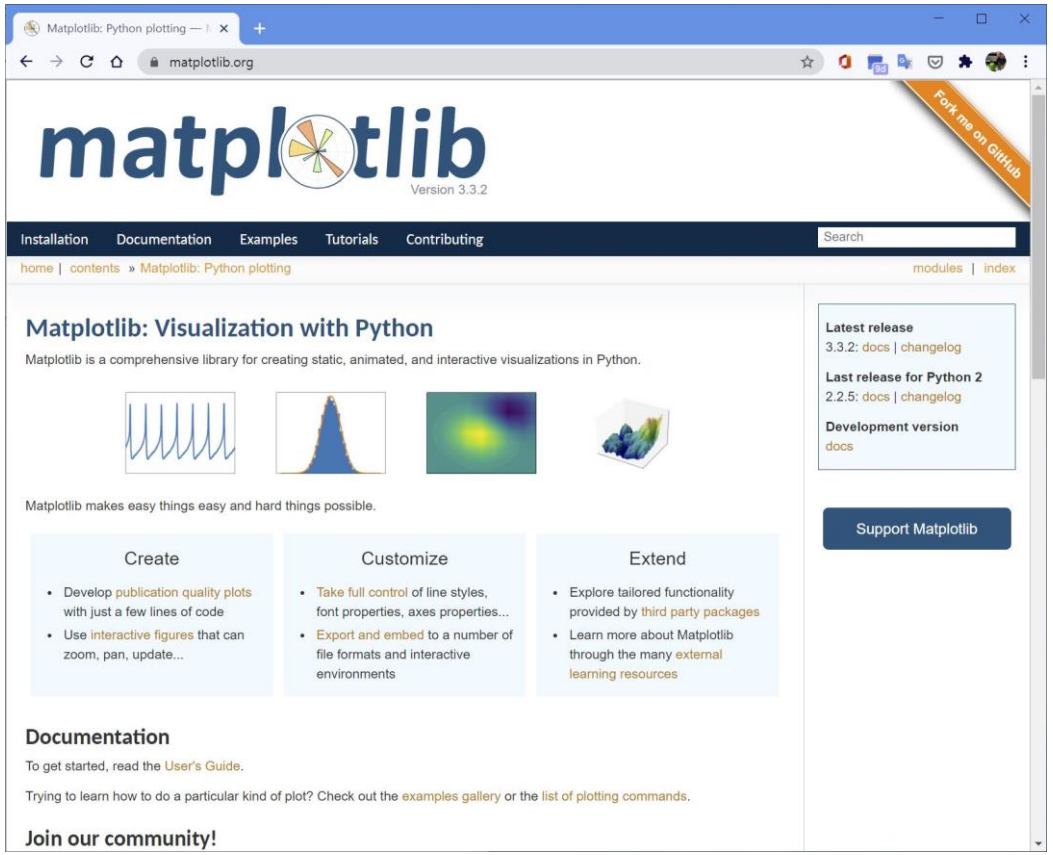
Jupyter Notebook

- Jupyter notebook은 대화형 파이썬 인터프리터(Interpreter)로서 웹 브라우저 환경에서 파이썬 코드를 작성 및 실행할 수 있는 도구
- 사용법 참고 : https://hogni.tistory.com/29



데이터 시각화 기초 : matplotlib

- 공식 사이트 : https://matplotlib.org
- Toturial : https://github.com/rougier/matplotlib-tutorial



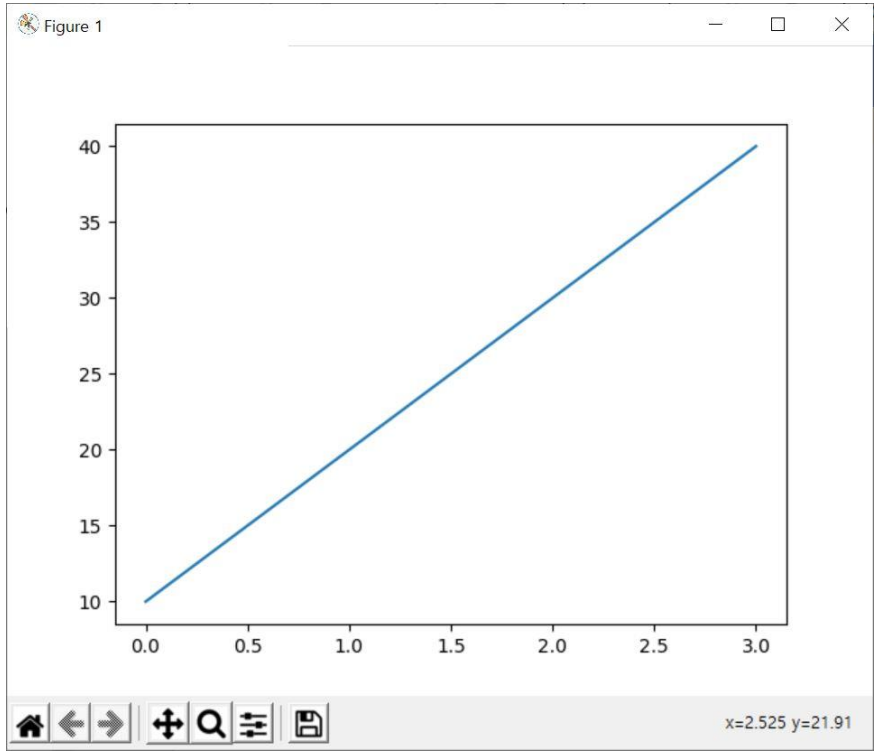
pyplot 모듈 불러오기

- mathplotlib 라이브러리에는 다양한 모듈들이 있는데, 우리는 그중 pyplot 모듈을 주로 사용함
  - https://matplotlib.org/py-modindex.html
- import matplotlib.pyplot
- import matplotlib.pyplot as plt

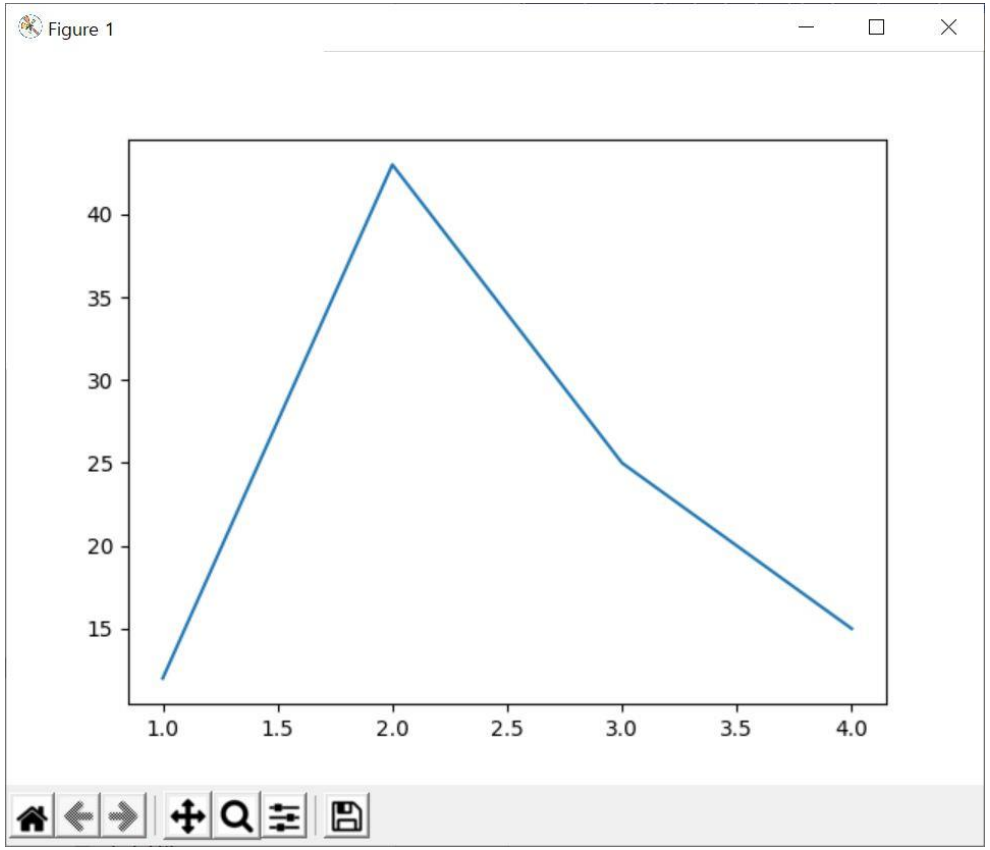
- from matplotlib import pyplot
- from matplotlib import pyplot as plt

기본 그래프 그리기

```
import matplotlib.pyplot as plt
plt.plot([10, 20, 30, 40])
plt.show()
```



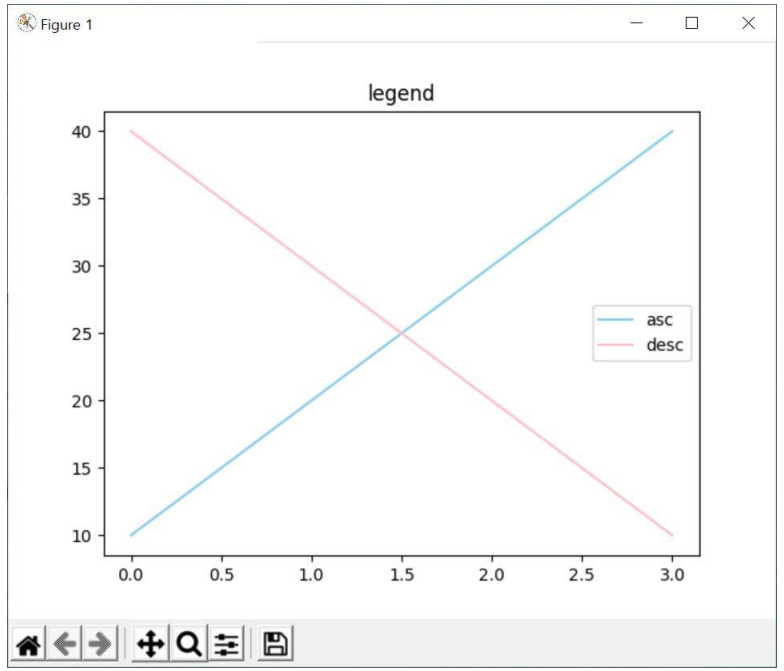
```python
from matplotlib import pyplot as plt
# plot([x축 데이터], [y축 데이터])
plt.plot([1,2,3,4], [12, 43, 25, 15])
plt.show()
```



그래프에 옵션 추가하기

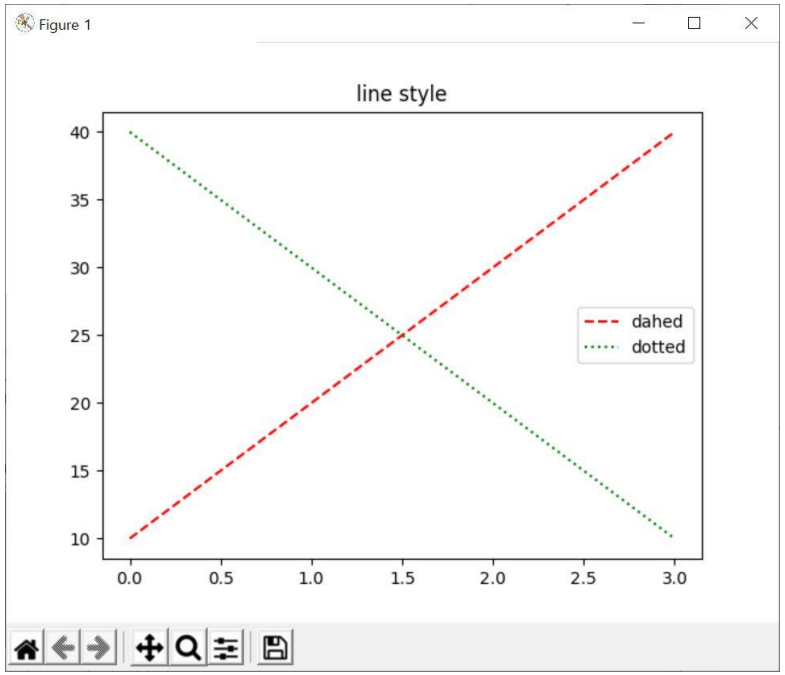
- 제목, 범례, 선 색깔

```
import matplotlib.pyplot as plt
plt.title('legend')
plt.plot([10, 20, 30, 40], color='skyblue', label='asc')
plt.plot([40, 30, 20, 10], 'pink', label='desc')
plt.legend()  # 범례 표시
plt.show()
```



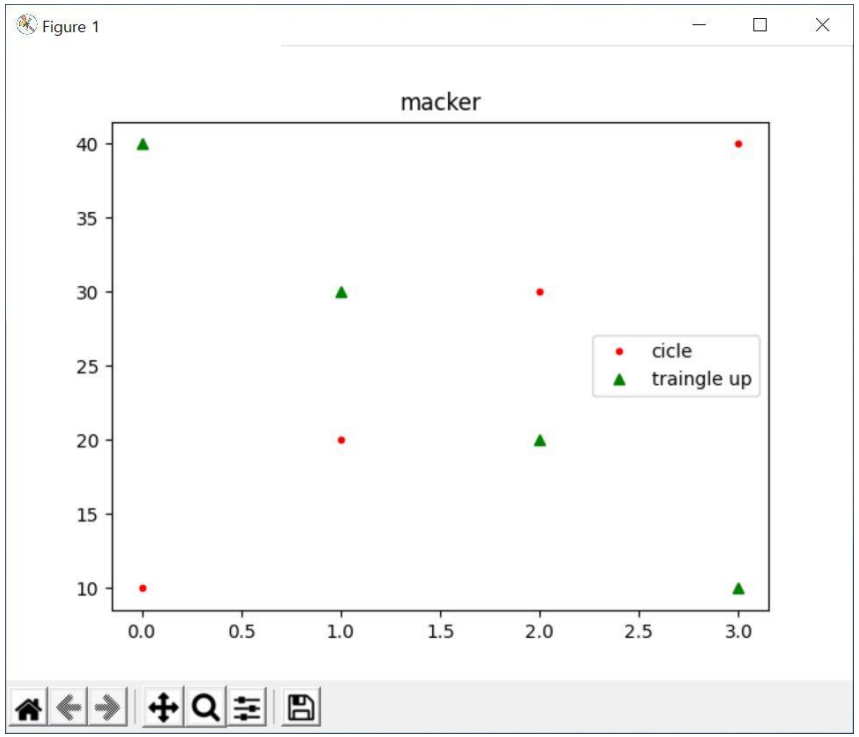
- 선 모양 바꾸기

```
import matplotlib.pyplot as plt
plt.title('line style')  # 제목 설정

# 빨간색 dashed 그래프
plt.plot([10, 20, 30, 40], color='r', linestyle='--', label='dahed')
#초록색 dotted 그래프
plt.plot([40, 30, 20, 10], color='g', ls=':', label='dotted')
plt.legend()  # 범례 표시
plt.show()
```



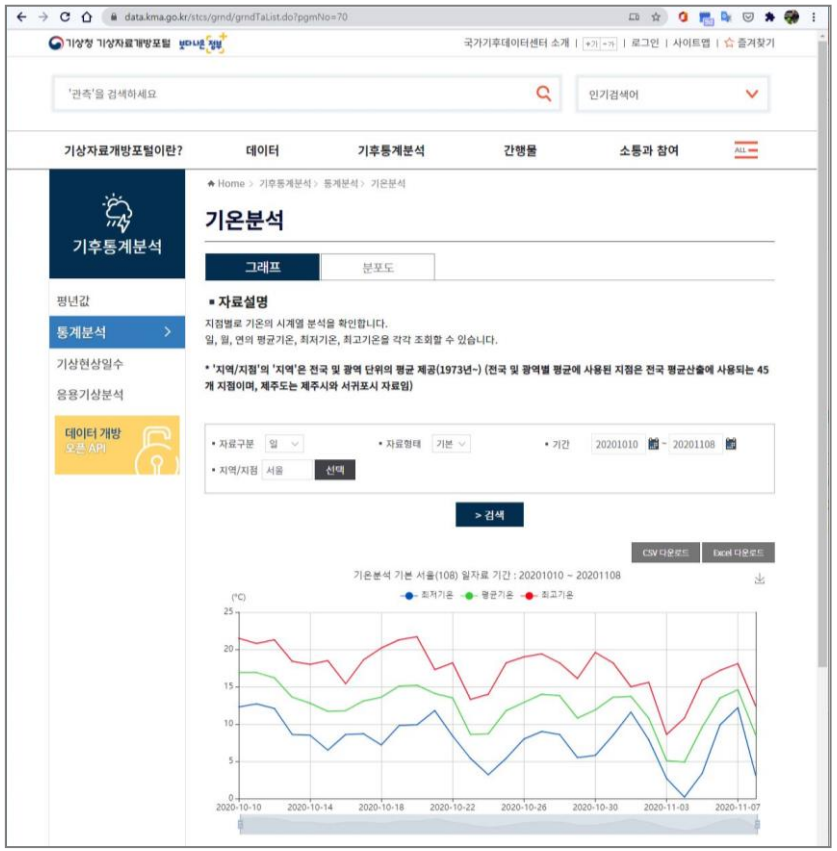
- 마커 모양 바꾸기
  - <색상>, <마커모양>, <선모양> 순으로 코드 작성

```
import matplotlib.pyplot as plt
plt.title('macker')  # 제목 설정

# 빨간색 원형 마크 그래프
plt.plot([10, 20, 30, 40], 'r.', label='cicle')

# 초록색 삼각형 마커 그래프
plt.plot([40, 30, 20, 10], 'g^', label='traingle up')

plt.legend()  # 범례 표시
plt.show()
```



부산의 기온 데이터 분석하기

- 기상청 기상자료개방포털
  - https://data.kma.go.kr/stcs/grnd/grndTaList.do?pgmNo=70



데이터 읽기/ 출력

- 데이터를 읽어 오기 전에 csv 파일 내용 중 필요 없는 필드는 삭제한다.

```
import csv
f = open('busan.csv', 'r', encoding='cp949')
data = csv.reader(f)
for row in data:
    print(row)
f.close()
```

```
['날짜', '지점', '평균기온(℃)', '최저기온(℃)', '최고기온(℃)']
['1904-07-01', '159', '23.7', '22.2', '27.3′]
['1904-07-02', '159', '21.8', '18.5', '23.6′]
['1904-07-03', '159', '18.8', '17.5', '21.8′]
['1904-07-04', '159', '20.6', '16.8', '25.2′]
['1904-07-05', '159', '24', '21.2', '28.2′]
['1904-07-06', '159', '24.9', '21.7', '28.5′]
['1904-07-07', '159', '25.3', '23.2', '28.4′]
['1904-07-08', '159', '25.5', '23.3', '29.2′]
['1904-07-09', '159', '25.6', '23.6', '31.5′]
['1904-07-10', '159', '20.6', '19.6', '24.6′]
['1904-07-11', '159', '19.1', '17.6', '20.1′]
['1904-07-12', '159', '19', '18.4', '20.1′]
['1904-07-13', '159', '19', '17.5', '19.9′]
['1904-07-14', '159', '21.3', '18.7', '23.6′]
 ['1904-07-15', '159', '23.7', '20.2', '27.6']
[생략]
```

헤더 저장하기

```
f = open('busan.csv', 'r', encoding='cp949')
data = csv.reader(f)
header = next(data)
#print(header)

for row in data:
    print(row)
f.close()
```

```
['1904-07-01', '159', '23.7', '22.2', '27.3′]
['1904-07-02', '159', '21.8', '18.5', '23.6′]
['1904-07-03', '159', '18.8', '17.5', '21.8′]
['1904-07-04', '159', '20.6', '16.8', '25.2′]
['1904-07-05', '159', '24', '21.2', '28.2′]
['1904-07-06', '159', '24.9', '21.7', '28.5′]
['1904-07-07', '159', '25.3', '23.2', '28.4′]
['1904-07-08', '159', '25.5', '23.3', '29.2′]
['1904-07-09', '159', '25.6', '23.6', '31.5′]
['1904-07-10', '159', '20.6', '19.6', '24.6′]
['1904-07-11', '159', '19.1', '17.6', '20.1′]
['1904-07-12', '159', '19', '18.4', '20.1′]
['1904-07-13', '159', '19', '17.5', '19.9′]
['1904-07-14', '159', '21.3', '18.7', '23.6′]
['1904-07-15', '159', '23.7', '20.2', '27.6']
[생략]
```

- 파이썬 csv 라이브러리 참고 : https://woolbro.tistory.com/35

부산의 최저 기온과 최고 기온 구하기

```
import csv
f = open('Busan.csv', 'r', encoding='cp949')
data = csv.reader(f)
header = next(data)

max_temp = -999
max_date = ''

for row in data:
    # 누락된 필드에 -999 추가
    if row[-1] == '':
        row[-1] = -999
    row[-1] = float(row[-1])

    if max_temp < row[-1] :
        max_date = row[0]
        max_temp = row[-1]
f.close()
print('부산의 기온이 최고인 날은', max_date+'로', max_temp,'도 였습니다.')
```

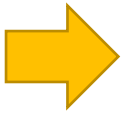
output : 부산의 기온이 최고인 날은 2016-08-14로 37.3 도 였습니다.

실습 01

- 부산의 기온이 가장 높았던 날/ 기온과 가장 낮았던 날/ 기온을 동시에 출력하는 프로그램을 작성하시오.
  - 기상관측 이래 현재까지의 데이터를 사용 바람
    - 1985 ~ 2020년
    - max_temp, max_date
    - min_temp, min_date

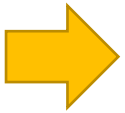
날짜 데이터 추출하기

```
s = ‘hello python’
print(s.split())
```



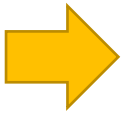
```
[‘hello’, ‘python’]
```

```
date = ‘1970-10-01’
print(date.split(‘-’)
```



```
[‘1970’, ‘10’, ‘01’]
```

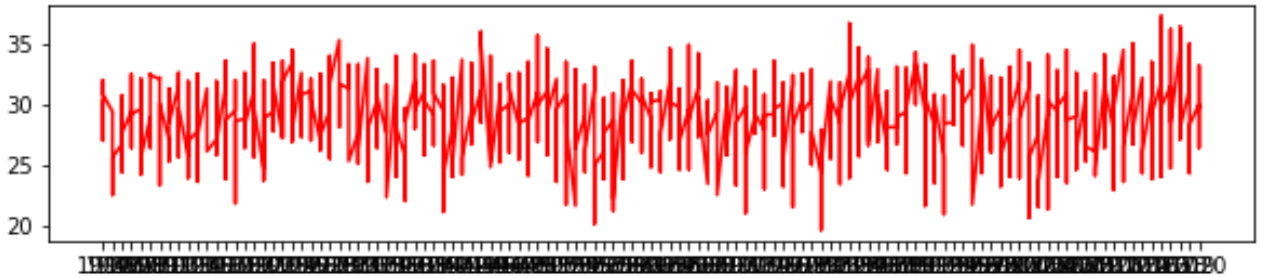
```
print(data.split(‘-’)[0])
print(data.split(‘-’)[1])
print(data.split(‘-’)[2])
```



```
1907
10
01
```

1년 중 여름의 정점인 8월의 최고기온?

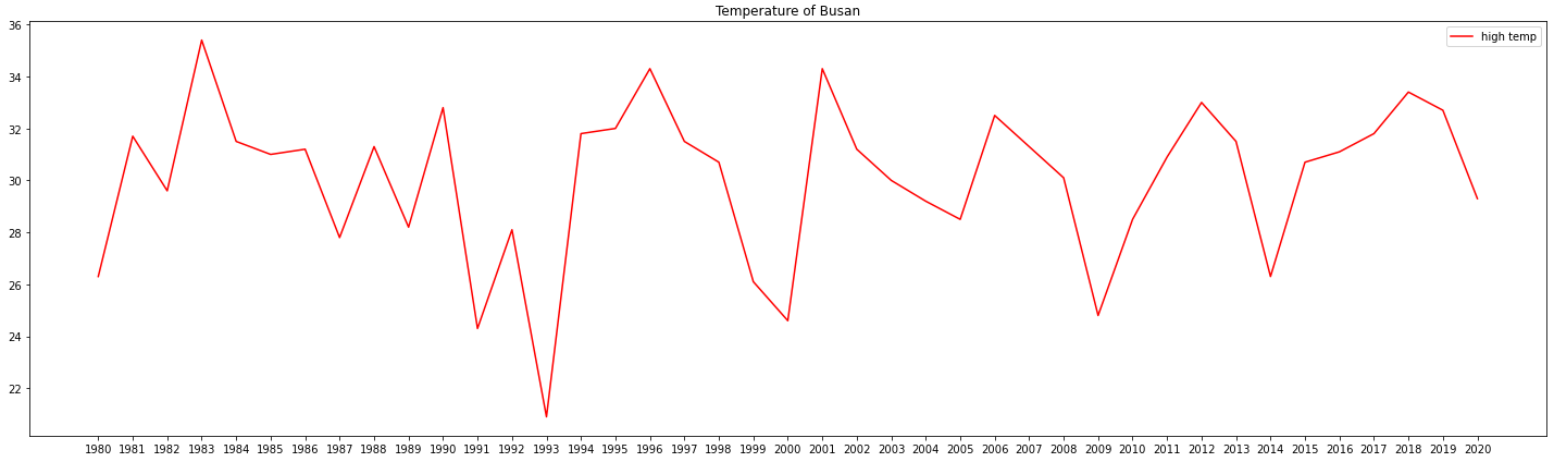
```
import matplotlib.pyplot as plt
import csv

f = open('Busan.csv', 'r', encoding='cp949')
data = csv.reader(f)
header = next(data)
result = []
year = []

for row in data :
    if row[-1] != '' :
        if row[0].split('-')[1] == '08':
            result.append(float(row[-1]))
            year.append(row[0].split('-')[0])

print(len(result))
plt.figure(figsize = (10, 2))
plt.plot(year, result, 'r')
plt.show()
f.close()
```



- matplotlib 그래프의 색상 이름 확인
  - http://bit.ly/2U5toVX

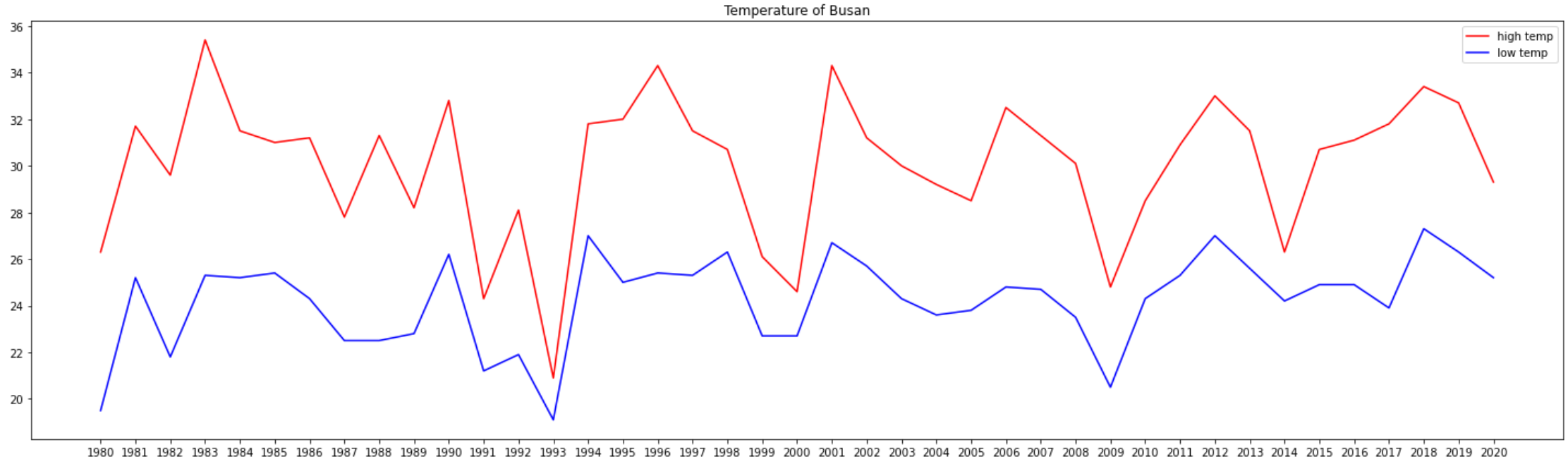
8월 10일 최고 기온은?

```
import matplotlib.pyplot as plt
import csv

f = open('Busan.csv', 'r', encoding='cp949')
data = csv.reader(f)
header = next(data)
high = [] # 최저 온도
year = []

for row in data :
    if row[-1] != '' and row[-2] != '' :
        if 1980 <= int(row[0].split('-')[0]) :
            if row[0].split('-')[1] == '08' and row[0].split('-')[2] == '03' :
                high.append(float(row[-2]))
                year.append(row[0].split('-')[0])
plt.figure(figsize = (25, 7))
plt.title('Temperature of Busan')
plt.plot(year, high, 'b', label='low temp')
plt.legend()
plt.show()
f.close()
```



실습 02

- 8월 10일의 최고 기온(high)와 최저 기온(low)를 동시에 플로팅 하는 프로그램을 작성 하시오.



과제

- 실습 01, 실습 02 구현하기
- 구글 클래스 룸
- 기한 : 2020. 11. 17 까지