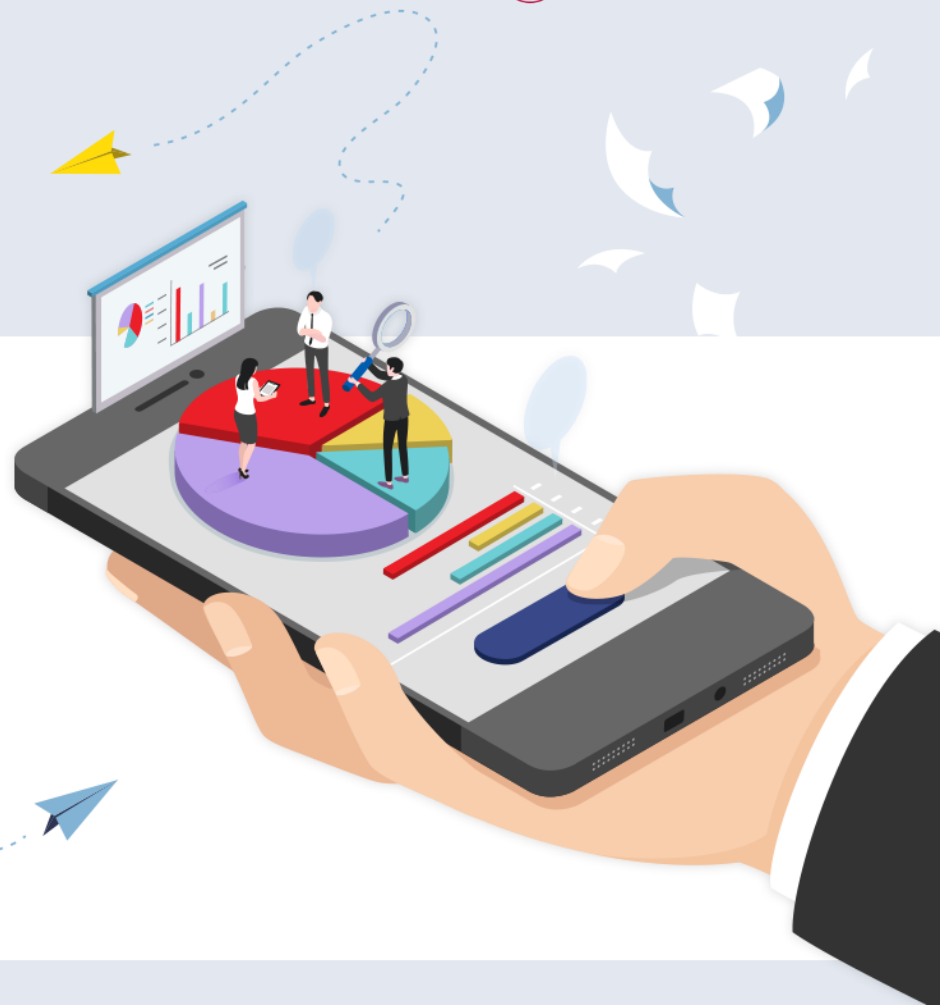




01

# 기온 공공 데이터 분석 시작하기



# 1 기온 공공 데이터 분석 시작하기



<https://data.kma.go.kr>

기상청 날씨데이터 서비스

## 기상자료개방포털

'관측'을 검색하세요



인기검색어



1

기상자료개방포털이란?

데이터

기후통계분석

간행물

소통과 참여



### > 평년값

- 우리나라 기후평년값
- 세계기후평년값

2

### > 통계분석

- 조건별통계
- 기온분석
- 강수량분석
- 다중지점통계
- 전운량 계급별일수
- 강수 계급별일수
- 바람장미

### > 기상현상일수

- 강수일수
- 황사일수
- 폭염일수
- 열대야일수
- 24절기
- 순위값
- 장마

### > 응용기상분석

- 체감온도(겨울철)
- 실효습도
- 열지수
- 냉/난방도일
- 적산온도



뜨거운 열기로 대한민국이 들썩였던  
한일 월드컵이 열린 **2002년도**의  
**서울의 기온**을 살펴볼까요?

# 1 기온 공공 데이터 분석 시작하기



분류, 지역/지점, 기간, 조건 등 설정을 마친 후 검색 버튼을 누름

▪ 분류

지상 ▾

▪ 지역/지점

서울

선택

▪ 요소

기온 ▾

▪ 기간

일 ▾

2002 ▾

년 ~

2002 ▾

년

▪ 조건

☐ 요소

평균기온 ▾

< ▾

선택 ▾

☒ 월

06 ▾

~

06 ▾

월

☒ 일

01 ▾

~

30 ▾

일

☐ 계절

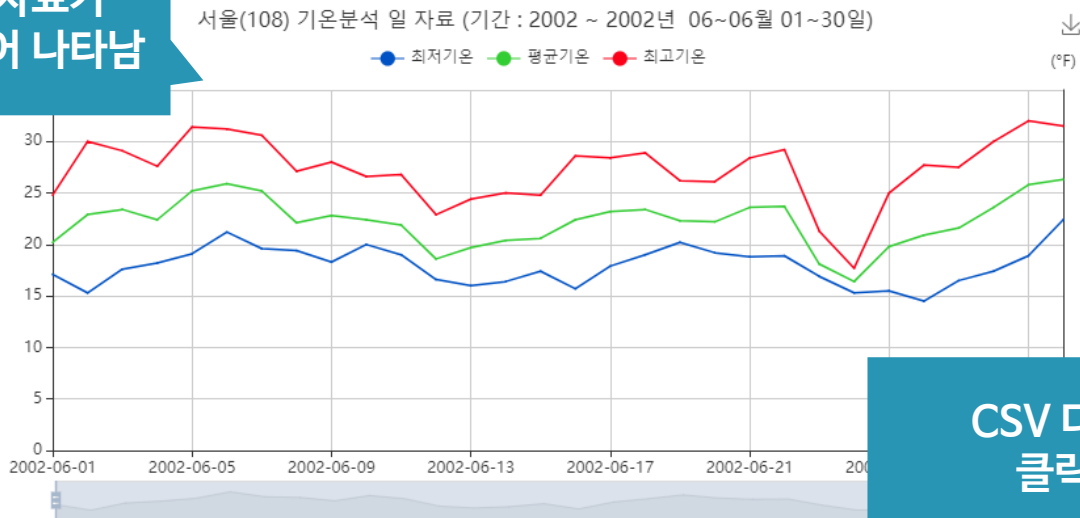
봄 ▾

&gt; 검색

# 1 기온 공공 데이터 분석 시작하기



조건에 맞게 검색된 자료가  
그래프와 표로 정리되어 나타남



CSV 다운로드 버튼을  
클릭하여 저장함

CSV 다운로드

Excel 다운로드

지점	지점명	일시	평균기온(°C)	최고기온(°C)	최고기온시각	최저기온(°C)	최저기온시각
108	서울	2002-06-01	20.2	24.8	12:29	17.1	00:06
108	서울	2002-06-02	22.9	30.0	15:08	15.3	04:50

## CSV의 개념

### CSV

- Comma-Seperated Values
- 각각의 데이터 값을 콤마(,)로 구분하는 파일 형식



정부에서 운영하는 공공데이터포털이 제공하는 일반적인 파일 형식으로 데이터 분석 전문가들이 자주 사용함



## CSV의 특징

- ✓ 더블 클릭하면 엑셀 프로그램에서 열 수 있고, 엑셀 파일처럼 사용할 수 있음
- ✓ 엑셀 프로그램이 없다면 메모장 같은 텍스트 편집기로 CSV파일을 열어 수정 및 생성할 수 있음

## [ 엑셀 ]

	A	B	C	D	E	F	G	H	
1									
2	계급별통계								
3	[검색조건]								
4		분류: 지상							
5		요소 : 기온							
6		검색구분 : 일							
7		지역/지점명 : 서울							
8		기간 : 2002 ~ 2002년 06 ~ 06월 01 ~ 30일							
9									
10									
11									
12	지점번호	지점명	일시	평균기온(°C)	최고기온(°C)	최고기온시간	최저기온(°C)	최저기온시간	
13	108	서울	2002-06-01	20.2	24.8	12:29	17.1	0:06	
14	108	서울	2002-06-02	22.9	30	15:08	15.3	4:50	
15	108	서울	2002-06-03	23.4	29.1	14:45	17.6	5:31	
16	108	서울	2002-06-04	22.4	27.6	14:06	18.2	4:29	
17	108	서울	2002-06-05	25.2	31.4	15:13	19.1	4:43	
18	108	서울	2002-06-06	25.9	31.2	13:07	21.2	5:57	
19	108	서울	2002-06-07	25.2	30.6	13:23	19.6	23:56	
20	108	서울	2002-06-08	22.1	27.1	17:03	19.4	0:15	
21	108	서울	2002-06-09	22.8	28	13:12	18.3	5:12	
22	108	서울	2002-06-10	22.4	26.6	13:23	20	23:00	
23	108	서울	2002-06-11	21.9	26.8	15:14	19	4:20	
24	108	서울	2002-06-12	18.6	22.9	10:47	16.6	4:08	
25	108	서울	2002-06-13	19.7	24.4	14:35	16	5:09	
26	108	서울	2002-06-14	20.4	25	13:00	16.4	5:18	
27	108	서울	2002-06-15	20.6	24.8	17:10	17.4	4:53	
28	108	서울	2002-06-16	22.4	28.6	15:46	15.7	5:06	
29	108	서울	2002-06-17	23.2	28.4	13:50	17.9	5:28	
30	108	서울	2002-06-18	23.4	28.9	14:08	19	5:56	
31	108	서울	2002-06-19	22.3	26.2	14:51	20.2	2:03	
32	108	서울	2002-06-20	22.2	26.1	14:42	19.2	8:38	

## [ 메모장 ]



계급별통계

[검색조건]

,분류: 지상

,요소 : 기온

,검색구분 : 일

,지역/지점명 : 서울

,기간 : 2002 ~ 2002년 06 ~ 06월 01 ~ 30일

지점번호,지점명,일시,평균기온(°C),최고기온(°C), 최고기온시간,최저기온(°C),최저기온시간

108,서울,2002-06-01,20.2,24.8,12:29,17.1,00:06

108,서울,2002-06-02,22.9,30.0,15:08,15.3,04:50

108,서울,2002-06-03,23.4,29.1,14:45,17.6,05:31

108,서울,2002-06-04,22.4,27.6,14:06,18.2,04:29

108,서울,2002-06-05,25.2,31.4,15:13,19.1,04:43

108,서울,2002-06-06,25.9,31.2,13:07,21.2,05:57



# 1 기온 공공 데이터 분석 시작하기



## CSV 파일 정제하기

2002.csv - Excel (제품 인증 실패)

삭제

	A	B	C	D	E	F	G	H	I
1									
2		계급별통계							
3		[검색조건]							
4		분류: 지상							
5		요소: 기온							
6		검색구분: 일							
7		지역/지점명: 서울							
8		기간: 2002 ~ 2002년 06 ~ 06월 01 ~ 30일							
9									
10									
11									
12	지점번호	지점명	일시	평균기온(°C)	최고기온(°C)	최저기온(°C)	최저기온(°C)	최저기온(°C)	최저기온(°C)
13	108	서울	2002-06-01	20.2	24.8	12:29	17.1	0:06	
14	108	서울	2002-06-02	22.9	30.0	15:08	15.3	4:50	
15	108	서울	2002-06-03	23.4	29.1	14:45	17.6	5:31	
16	108	서울	2002-06-04	22.4	27.6	14:06	18.2	4:29	
17	108	서울	2002-06-05	25.2	31.4	15:13	19.1	4:43	

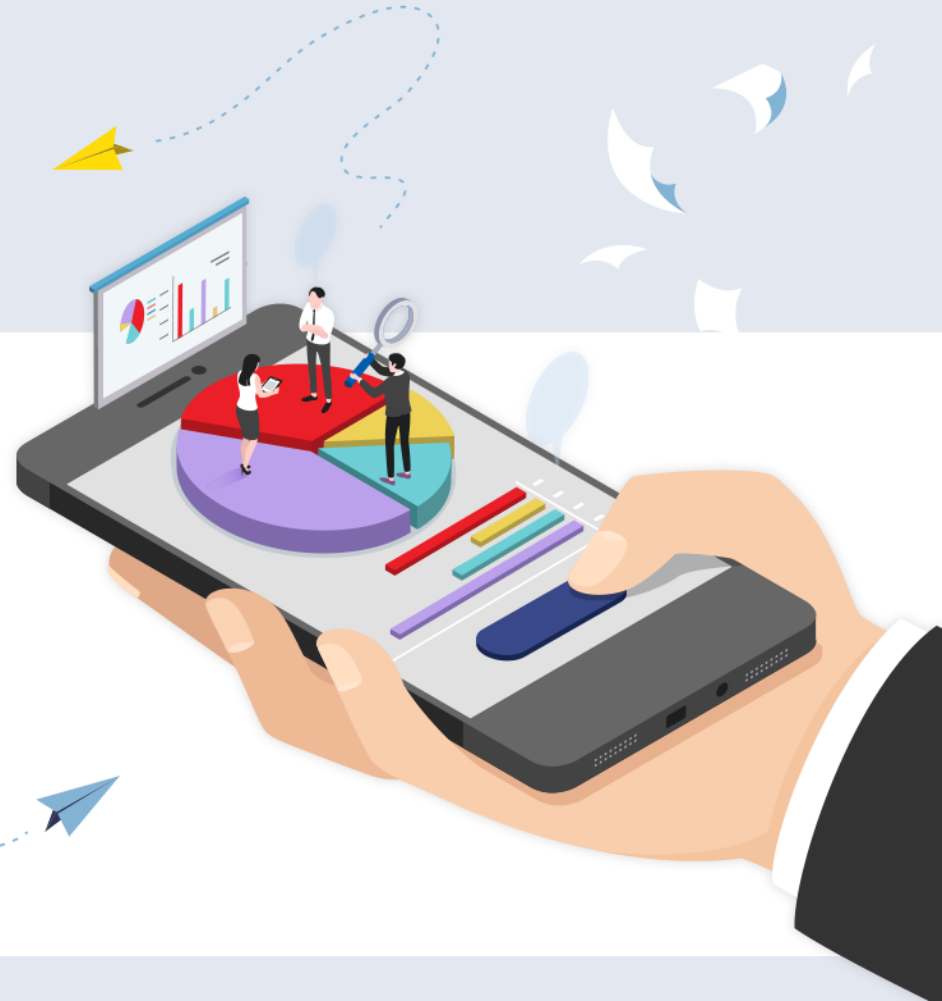
이 과정에서 경고창이 떠도  
당황하지 말고 예(Y) 버튼을 누르기

파일 이름(N): 2002.csv

파일 형식(T): CSV (쉼표로 분리) (\*.csv)



# 02 데이터 시각화



## SETP 1 : CSV 파일에서 데이터 읽어오기

<code>import csv</code>	<b>#1 csv 모듈을 불러옴</b>
<code>f=open('2002.csv')</code>	<b>#2 csv 파일을 open() 함수를 열어서 f에 저장함</b>
<code>data=csv.reader(f)</code>	<b>#3 csv 파일에서 데이터를 읽어와서 data에 저장함</b>
<code>header=next(data)</code>	<b>#4 header라는 변수에 헤더 데이터 행을 저장함</b>
<code>print(header)</code>	<b>#5</b>
<code>for row in data:</code>	<b>#6</b>
<code>    print(row)</code>	<b>#7</b>
<code>f.close()</code>	<b>#8</b>

## SETP 1 : CSV 파일에서 데이터 읽어오기

```
import csv                #1

f=open('2002.csv')        #2
data=csv.reader(f)         #3
header=next(data)         #4
print(header)             #5  header를 출력함
for row in data:           #6  데이터를 한 줄씩 읽어와서
    print(row)             #7  화면에 출력함
f.close()                 #8  파일을 닫음
```



## SETP 1 : CSV 파일에서 데이터 읽어오기

〈실행결과〉 ['일시', '평균기온(℃)', '최고기온(℃)', '최저기온(℃)']

['2002-06-01', '20.2', '24.8', '17.1']

['2002-06-02', '22.9', '30', '15.3']

['2002-06-03', '23.4', '29.1', '17.6']

['2002-06-04', '22.4', '27.6', '18.2']

['2002-06-05', '25.2', '31.4', '19.1']

['2002-06-06', '25.9', '31.2', '21.2']

['2002-06-07', '25.2', '30.6', '19.6']

['2002-06-08', '22.1', '27.1', '19.4']

['2002-06-09', '22.8', '28', '18.3']

['2002-06-10', '22.4', '26.6', '20']

['2002-06-11', '21.9', '26.8', '19']

...생략됨

## 💬 SETP 1 : CSV 파일에서 데이터 읽어오기

〈실행결과〉 ['일시', '평균기온(℃)', '최고기온(℃)', '최저기온(℃)']

['2002-06-01', '20.2', '24.8', '17.1']

['2002-06-02', '22.9', '30', '15.3']

['2002-06-03', '23.4', '29.1', '17.6']

['2002-06-04', '22.4', '27.6', '18.2']

['2002-06-05', '25.2', '31.4', '19.1']

['2002-06-06', '25.9', '31.2', '21.2']

['2002-06-07', '25.2', '30.6', '19.6']

['2002-06-08', '22.1', '27.1', '19.4']

['2002-06-09', '22.8', '28', '18.3']

실행결과 분석

- 각 행의 데이터는 리스트로 반환됨
- 각 행의 데이터가 문자열 데이터로 이루어져 있음



## SETP 2 : 평균기온데이터에 대한 시각화

```
import csv
```

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

```
f=open('2002.csv')
```

```
data=csv.reader(f)
```

```
header=next(data)
```

```
print(header)
```

```
temp= []
```

```
for row in data:
```

```
    temp.append(row[1])
```

```
temp=list(map(float,temp))
```

```
print(temp)
```

#1 matplotlib라이브러리에 속한 pyplot 모듈을 불러옴

#2 평균기온을 저장하기 위한 리스트 생성

#3 평균기온을 리스트에 저장

#4 평균기온 데이터는 숫자 값이 아닌 문자열이기때문에 값을 숫자 데이터형으로 변환

## 💬 SETP 2 : 평균기온데이터에 대한 시각화

### # 그래프 그리기

```
plt.title('Average temperature graph')
```

#1 title() 함수 - 그래프에 제목 넣기

```
plt.plot(temp,linewidth=5)
```

#2 plot() 함수는 직선 또는 꺾은 선 형태의 그래프를 그릴 때 사용

```
plt.xlabel('day')
```

#3 xlabel() 함수-x축 레이블지정/  
ylabel() 함수-y축 레이블지정

```
plt.ylabel('temperature')
```

```
plt.legend(['temperature'])
```

#4 legend() 함수-범례넣기

```
plt.show()
```

#5 show() 함수-그래프 보여주기

```
f.close()
```



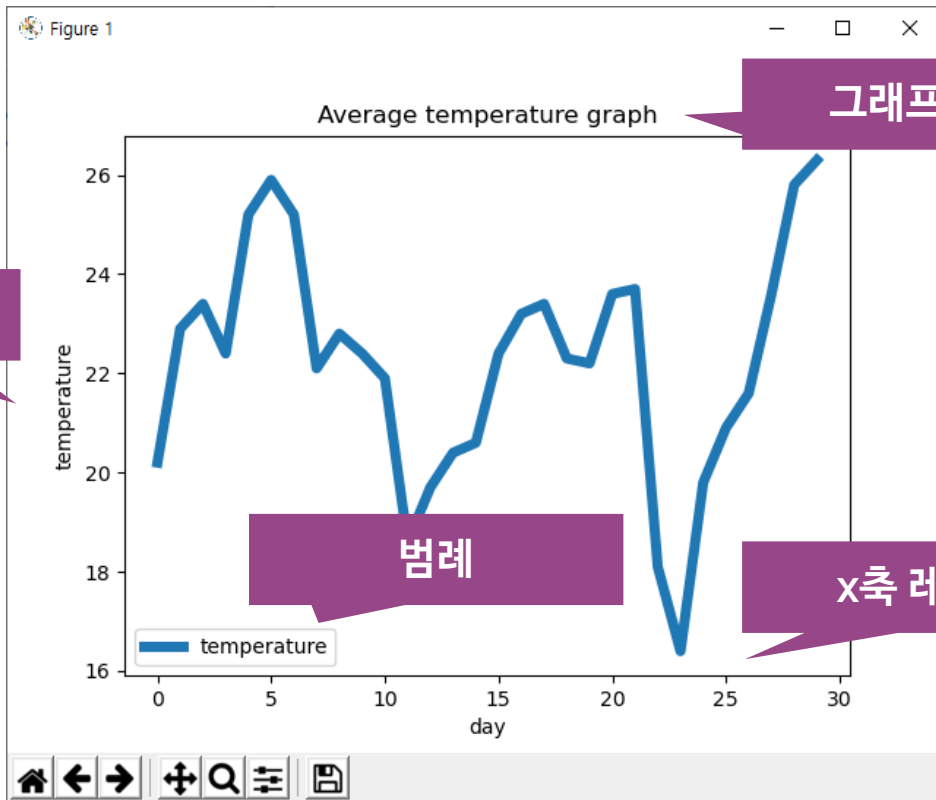
# 1 파이썬 코드로 구현하기



## SETP 2 : 평균기온데이터에 대한 시각화

### 〈실행결과〉

y축 레이블



그래프 제목

범례

x축 레이블



# 실습하기