

# Python 모듈



# 모듈(Module)

- 모듈

- 함수나 변수 또는 클래스 들을 모아 놓은 파일
- 다른 파이썬 프로그램에서 불러와 사용할 수 있게끔 만들어진 파이썬 파일

- 모듈 불러오기

- `import 모듈이름`
  - 모듈 이름은 .py 확장자를 제거한 파일이름
- `import 모듈이름 as 별명`
  - 모듈 이름을 별명으로 사용
- `from 모듈이름 import 모듈함수`
  - 모듈 이름을 붙이지 않고 바로 해당 모듈의 함수

- 대표적인 모듈

- `matplotlib.pyplot` : 시각화 모듈
- `numpy` : 수치해석 모듈
- `pandas` : 데이터 분석 모듈
- `requests` : HTTP 요청/응답 모듈



# 모듈 불러오기

## numcheck.py

```
#홀수
def is_odd(n):
    if n % 2 == 1 : return True
    else : return False

#짝수
def is_even(n):
    if n % 2 == 0 : return True
    else : return False

#소수
def is_prime(n) :
    for i in range(2, n) :
        if n % i == 0 : return False

    return True

#완전수 : 자기 자신을 제외한 모든 양의 약수들의 합이 자기 자신이 되는 자연수
def is_perfect(n):
    sum = 0
    for i in range(1,n):
        if n % i == 0 : sum = sum + i

    if sum == n : return True
    else : return False
```

```
if is_perfect(12) : print('완전수')
else : print("완전수아님")
```

```
if __name__ == '__main__' :
    if is_perfect(12) : print('완전수')
    else : print("완전수아님")
```

## num.py

```
import numcheck
```

```
if numcheck.is_perfect(6) : print('완전수')
else : print("완전수아님")
```

실행결과

완전수아님  
완전수

실행결과

완전수



# 데이터 시각화

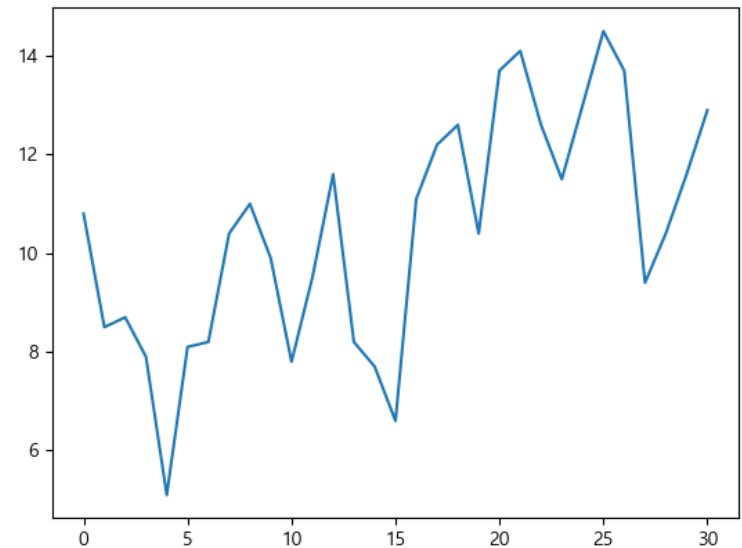
- 시각화 도구

- matplotlib.pyplot

```
#시각화 모듈
import matplotlib.pyplot as plt

#부산 3월 기온
temp = [10.8, 8.5, 8.7, 7.9, 5.1,
        8.1, 8.2, 10.4, 11, 9.9,
        7.8, 9.5, 11.6, 8.2, 7.7,
        6.6, 11.1, 12.2, 12.6, 10.4,
        13.7, 14.1, 12.6, 11.5, 13,
        14.5, 13.7, 9.4, 10.4, 11.6, 12.9]

plt.plot(temp)
plt.show()
```



# 데이터 시각화

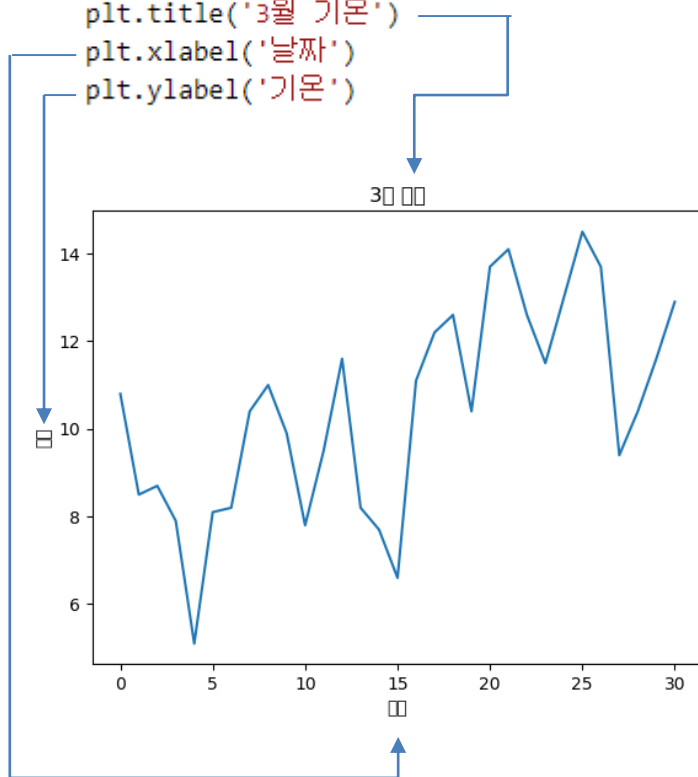
## • 제목 달기

#그래프 제목 달기

plt.title('3월 기온')

plt.xlabel('날짜')

plt.ylabel('기온')



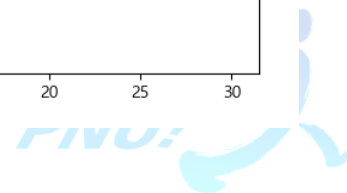
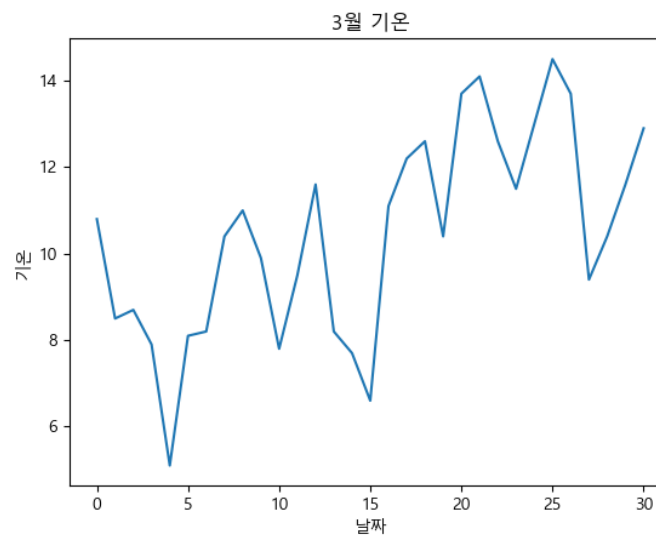
#한글 폰트 사용

from matplotlib import font\_manager, rc

font\_name =

font\_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get\_name()

rc('font', family=font\_name)



# 데이터 시각화

- 선 옵션

- 색상

- b(파란색), g(초록색), r(빨간색), c(청록색), y(노란색), k(검은색), w(흰색)

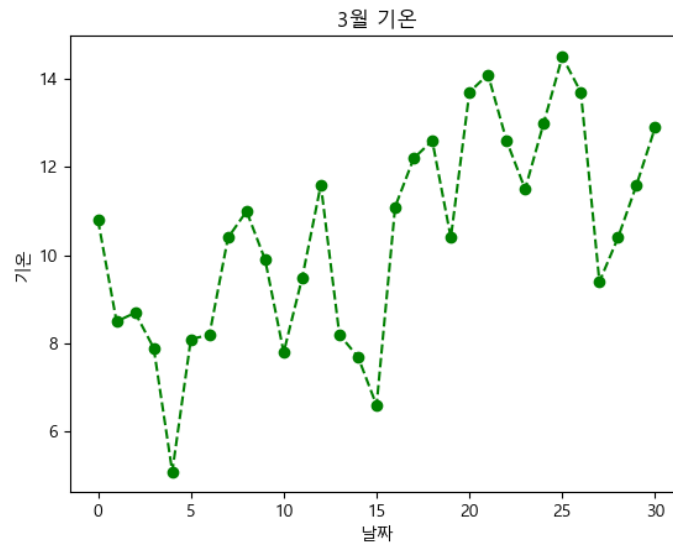
- 마커

- o(원), v(역삼각형), ^ (삼각형), s(네모), +(플러스), .(점)

- 선스타일

- -(solid line), --(dashed line ), -. (dash-dot line), : (dotted line )

```
#그래프 그리기  
plt.plot(temp, 'go--')
```



# 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



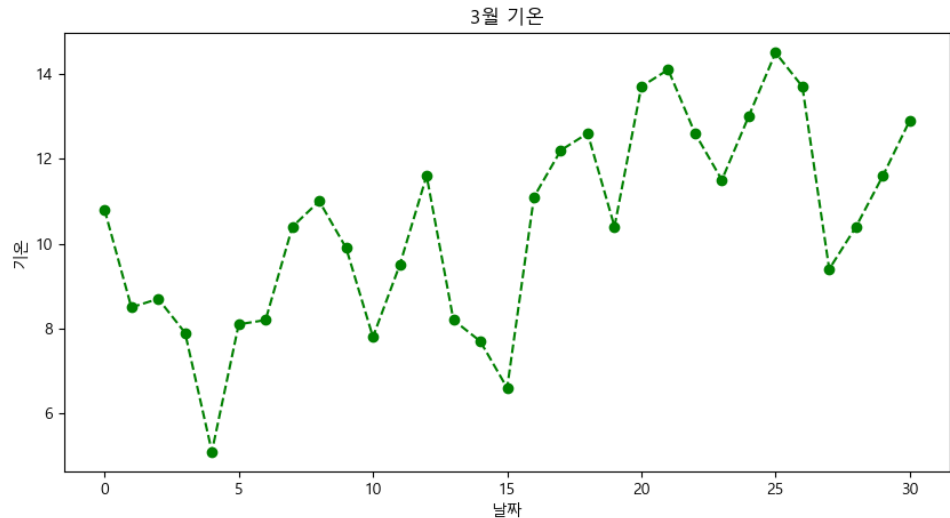
# 데이터 시각화

- 그래프 크기 조절

- plt.figure()

- figsize를 이용하여 가로, 세로 길이 조절 가능 (inch 단위)

```
#그래프 크기 조절: 가로 세로를 튜플로  
plt.figure(figsize=(10,5))
```



# 데이터 시각화

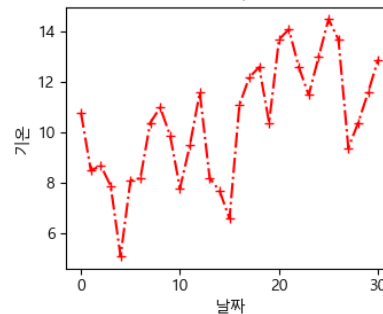
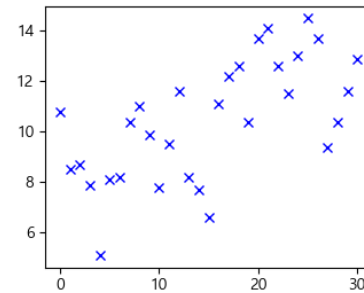
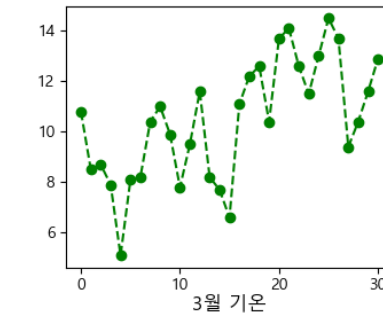
## • 여러 그래프 그리기1

```
#여러 그래프 그리기
#그래프 그리기
plt.subplot(2,2,1)
plt.plot(temp, 'go--')

plt.subplot(2,2,2)
plt.plot(temp, 'bx')
```

```
plt.subplot(2,2,3)
plt.plot(temp, 'r+-.')

#그래프 제목 달기
plt.title('3월 기온')
plt.xlabel('날짜')
plt.ylabel('기온')
```



# 데이터 시각화

## • 여러 그래프 그리기2

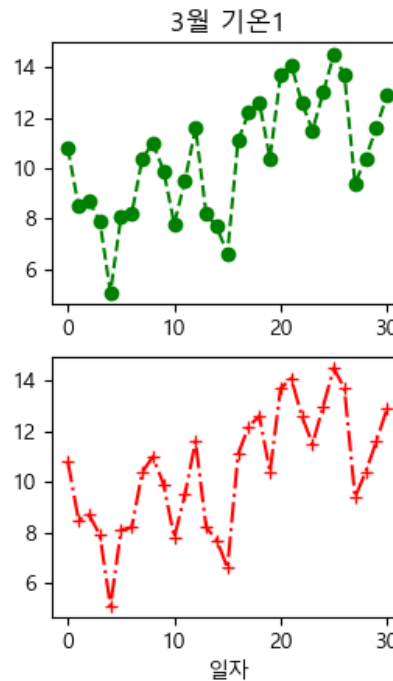
#여러 그래프 그리기 방법2

```
fig = plt.figure()  
ax1 = fig.add_subplot(2,2,1)  
ax2 = fig.add_subplot(2,2,2)  
ax3 = fig.add_subplot(2,2,3)
```

```
ax1.plot(temp, 'go--')  
ax1.set_title('3월 기온1')
```

```
ax2.plot(temp, 'bx')  
ax2.set_title('3월 기온2')
```

```
ax3.plot(temp, 'r+-.')  
ax3.set_xlabel('일자')
```



# 데이터 시각화

## • 그래프 겹치기

#그래프 겹쳐 그리기

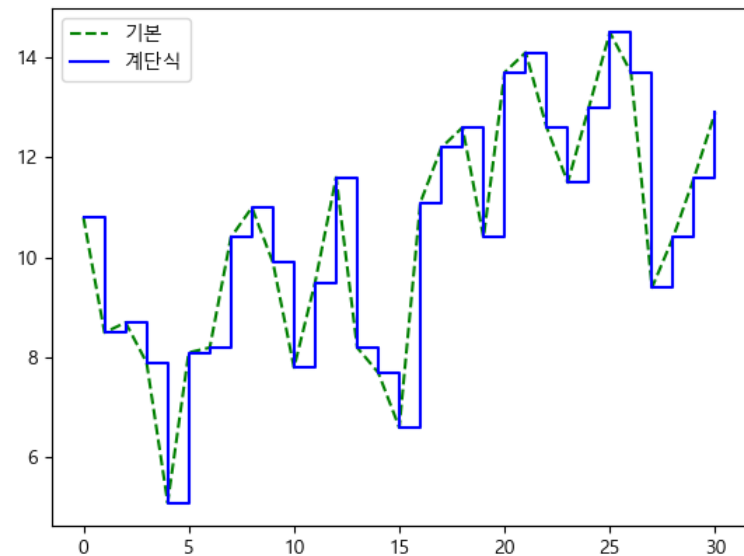
```
plt.plot(temp, 'g--', label='기본')
```

```
plt.plot(temp, 'b-', drawstyle='steps-post', label='계단식')
```

```
plt.legend(loc='best')
```

- 범례위치

best(0), upper right(1), upper left(2), lower left(3), lower right(4), right(5), center left(6), center right(7), lower center(8), upper center(9), center(10)



# 데이터 시각화

## • 그래프 종류

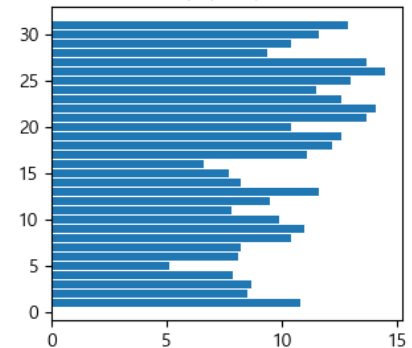
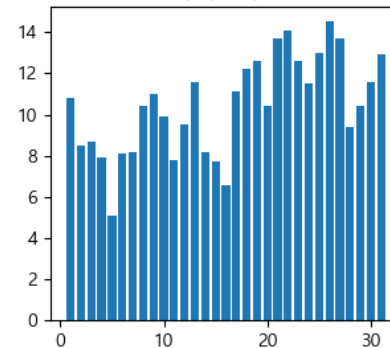
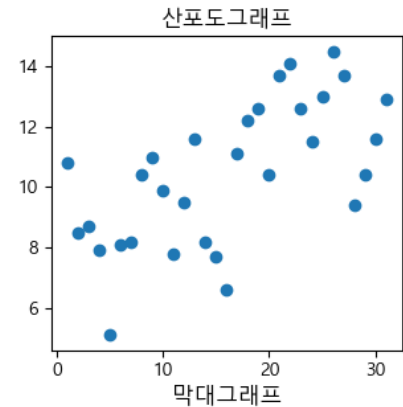
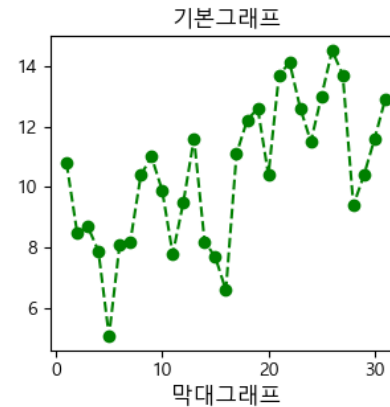
```
#여러 그래프 그리기
#그래프 그리기
x = list(range(1,32))

plt.subplot(2,2,1)
plt.plot(x,temp, 'go--')
plt.title('기본그래프')

plt.subplot(2,2,2)
plt.scatter(x, temp)
plt.title('산포도그래프')

plt.subplot(2,2,3)
plt.bar(x, temp)
plt.title('막대그래프')

plt.subplot(2,2,4)
plt.barh(x,temp)
plt.title('막대그래프')
```



# 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]
```

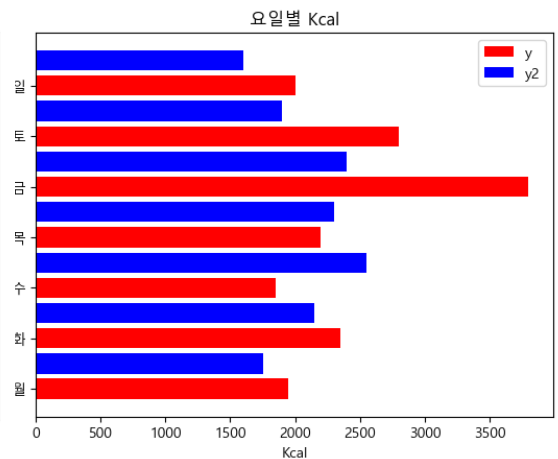
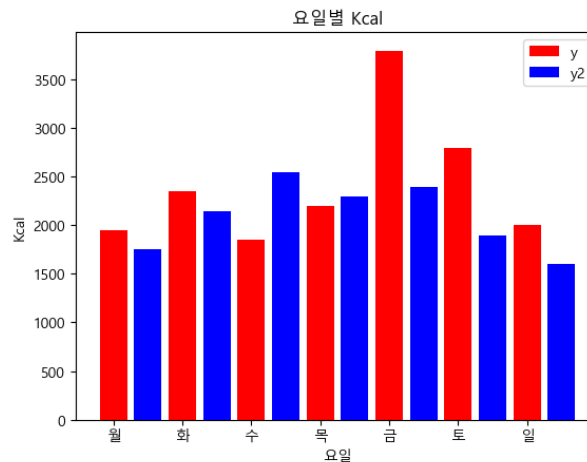
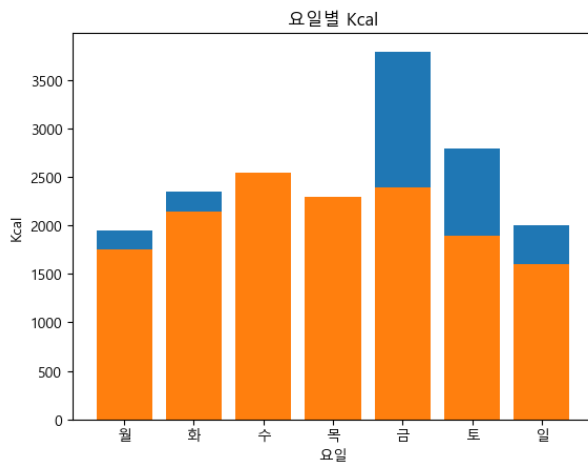
```
plt.bar(x, y)
plt.bar(x, y2)
plt.xticks(x, x2)
```

```
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)
```

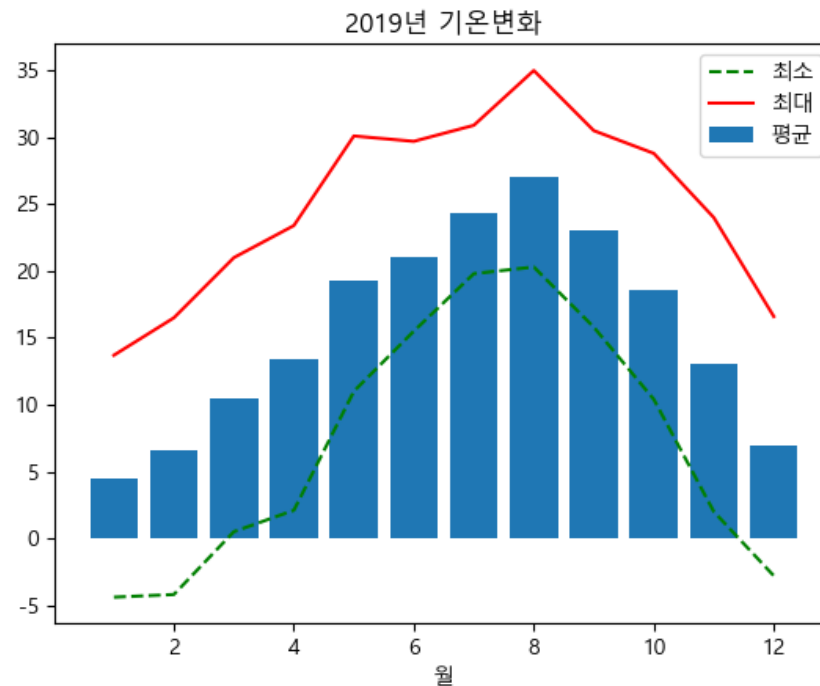
```
plt.barh(x, y, label='y', color='r')
plt.barh(x1, y2, label='y2', color='b')
plt.yticks(x, x2)
```

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



# 해결문제

- 2019년 부산의 1월에서 12월 평균, 최저, 최고 기온의 그래프를 아래와 같이 작성하시오.
  - 그래프에서 마이너스 기호가 표시  
`plt.rcParams['axes.unicode_minus'] = False`



# 해결문제

- 다음은 시도별 대기정보이다.

- [https://www.airkorea.or.kr/web/sidoQualityCompare?itemCode=10008&pMENU\\_NO=102](https://www.airkorea.or.kr/web/sidoQualityCompare?itemCode=10008&pMENU_NO=102)
- 시도별 대기정보는 미세먼지(PM10), 오존(O3), 일산화탄소(CO)의 정보가 순서대로 주어진다.

