

Python 함수/모듈



함수

- 특정 기능을 하나로 묶어서 따로 관리하기 위해 사용
 - 반복되는 내용을 효율적으로 처리
 - 코드의 가독성을 높임

- 함수 구조

```
def 함수명(인자1, 인자2 ...):  
    #함수 코드  
    return 리턴값
```

함수 매개변수와 반환값

반환 값 없음

```
1 def cal(num1, num2, op) :
2     ans = 0
3     if op == "+" : ans = num1 + num2
4     elif op == "-" : ans = num1 - num2
5     elif op == "*" : ans = num1 * num2
6     elif op == "/" : ans = num1 / num2
7
8     print(num1, op, num2, "=", ans)
9
10
11 if __name__ == "__main__" :
12     while(True) :
13
14         num1 = int(input("숫자를 입력하세요=>"))
15         num2 = int(input("숫자를 입력하세요=>"))
16         op = input("연산자를 입력하세요(+, -, *, /)=>")
17
18         if ( op not in ["+", "-", "*", "/"]): break
19
20         cal(num1, num2, op)
21
```

반환 값 1개

```
1 def cal(num1, num2, op) :
2     ans = 0
3     if op == "+" : ans = num1 + num2
4     elif op == "-" : ans = num1 - num2
5     elif op == "*" : ans = num1 * num2
6     elif op == "/" : ans = num1 / num2
7
8     return ans
9
10
11 if __name__ == "__main__" :
12     while(True) :
13
14         num1 = int(input("숫자를 입력하세요=>"))
15         num2 = int(input("숫자를 입력하세요=>"))
16         op = input("연산자를 입력하세요(+, -, *, /)=>")
17
18         if ( op not in ["+", "-", "*", "/"]): break
19
20         print(num1, op, num2, "=", cal(num1, num2, op))
21
```

반환 값 2개

```
1 def cal(num1, num2, op) :
2     ans = 0
3     if op == "+" : ans = num1 + num2
4     elif op == "-" : ans = num1 - num2
5     elif op == "*" : ans = num1 * num2
6     elif op == "/" : ans = num1 / num2
7
8     strans = str(num1) + op + str(num2) + "="
9     return strans, ans
10
11
12 if __name__ == "__main__" :
13     while(True) :
14
15         num1 = int(input("숫자를 입력하세요=>"))
16         num2 = int(input("숫자를 입력하세요=>"))
17         op = input("연산자를 입력하세요(+, -, *, /)=>")
18
19         if ( op not in ["+", "-", "*", "/"]): break
20
21         strans, ans = cal(num1, num2, op)
22         print(strans, ans)
```

해결문제

- 도별미세먼지.csv 파일을 읽어서 해당 지역의 미세먼지를 알려주는 함수를 작성하시오.
 - 단, 지역 입력에 q가 입력이 되면 종료

지역 선택

['대전', '전남', '서울', '경북', '제주', '강원', '울산', '충남', '인천', '광주', '세종', '경기', '전북', '충북', '경남', '대구', '부산']

지역을 입력하세요(종료 q) : 대전

대전 의 미세먼지 : 22

지역을 입력하세요(종료 q) : 대구

대구 의 미세먼지 : 24

지역을 입력하세요(종료 q) : 부산

부산 의 미세먼지 : 26

지역을 입력하세요(종료 q) : Q



모듈

- 파이썬 코드를 논리적으로 묶어서 관리하고 사용할 수 있도록 하는 것
 - 함수, 클래스, 혹은 변수들이 정의될 수 있으며, 실행 코드를 포함
- 표준 라이브러리 모듈
 - import 모듈명
 - import math
 - from 모듈명 import 함수명
 - from math import factorial



Python 시각화 모듈

- 시각화 도구

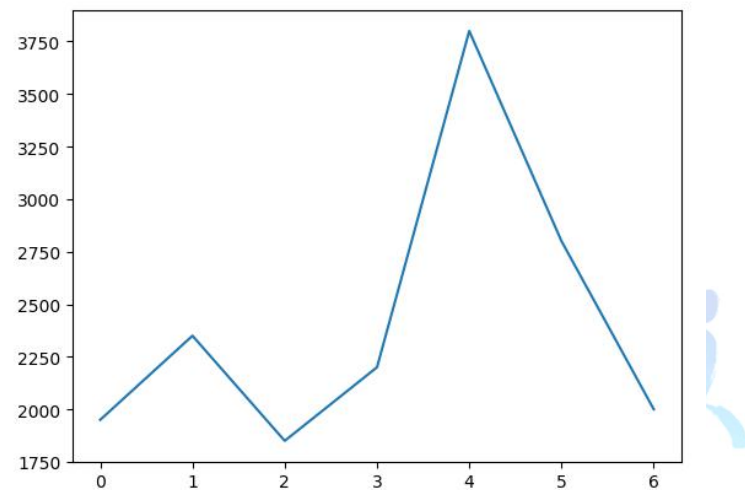
- matplotlib.pyplot

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

```
y = [1950, 2350, 1850, 2200, 3800, 2800, 2000]
```

```
plt.plot(y)
```

```
plt.show()
```



데이터 시각화

- X축 표기 및 한글화

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

```
#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)
```

```
x = [0,1,2,3,4,5,6]
```

```
x2 = ['월','화','수','목','금','토','일']
```

```
y = [1950,2350,1850, 2200,3800,2800,2000]
```

```
plt.plot(y)
```

```
plt.xticks(x, x2)
```

```
plt.show()
```



데이터시각화

• 옵션 및 레이블

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

```
#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)
```

```
x = [0,1,2,3,4,5,6]
```

```
x2 = ['월','화','수','목','금','토','일']
```

```
y = [1950,2350,1850, 2200,3800,2800,2000]
```

```
plt.plot(y, 'bo-')
```

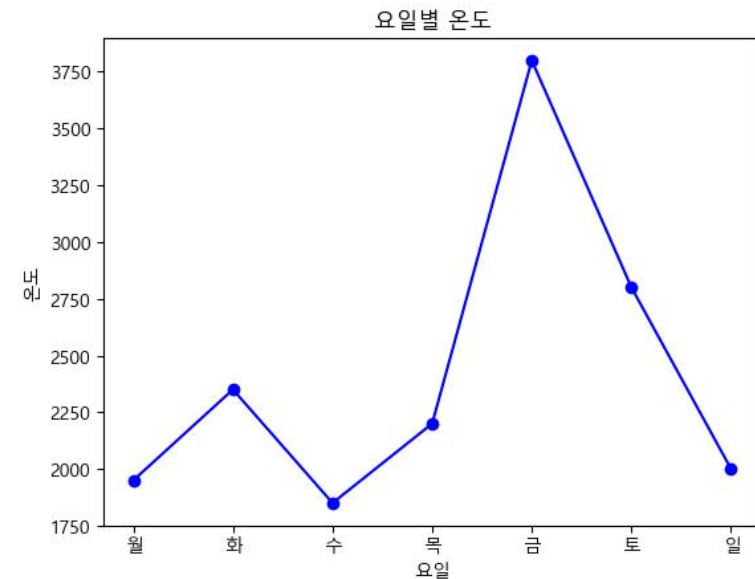
```
plt.xticks(x, x2)
```

```
plt.xlabel('요일')
```

```
plt.ylabel('온도')
```

```
plt.title('요일별 온도')
```

```
plt.show()
```



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

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

```
#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)
```

```
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.plot(x, y, label='y', c='r', lw=4, ls=':', marker='x')
```

```
plt.plot(x, y2, label='y2', c='b', lw=2, ls='--', marker='o')
```

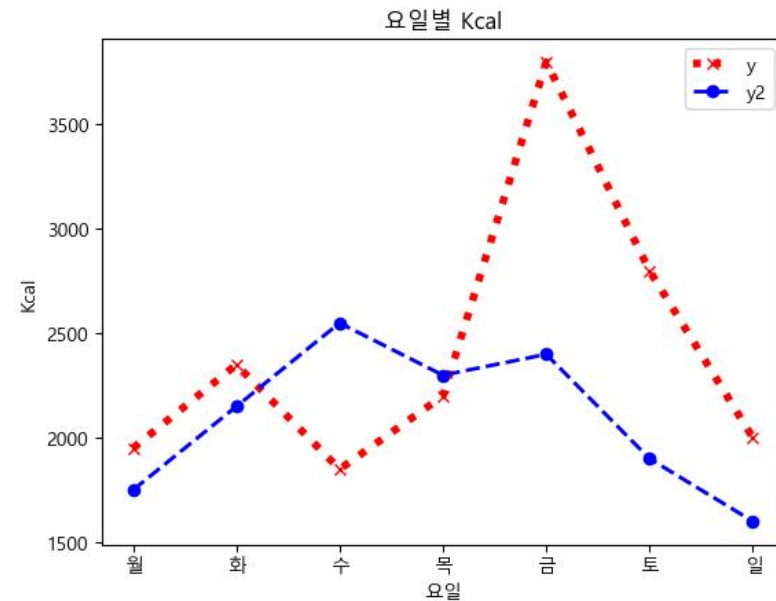
```
plt.xticks(x, x2)
```

```
plt.xlabel('요일')
```

```
plt.ylabel('온도')
```

```
plt.title('요일별 온도')
```

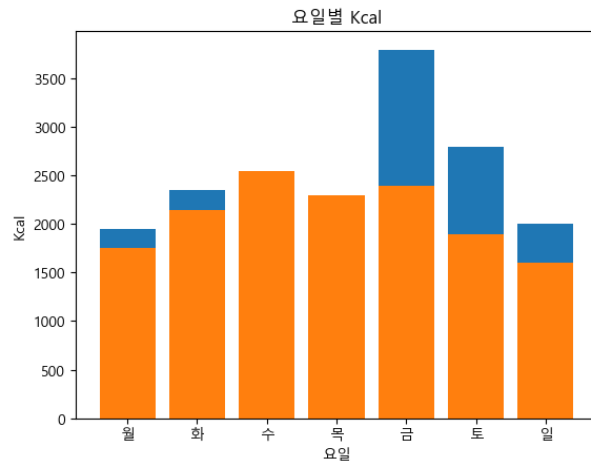
```
plt.show()
```



데이터시각화

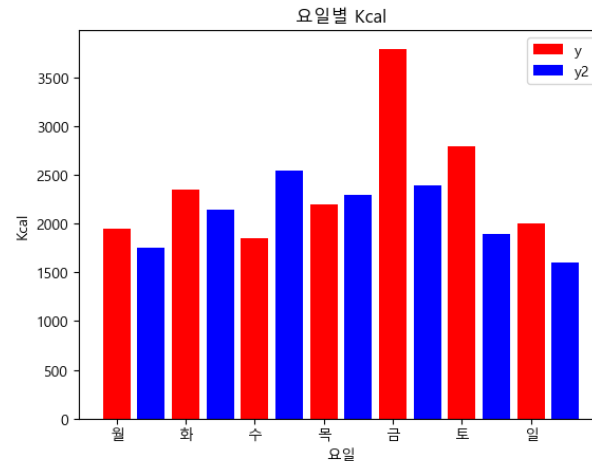
```
x = [0,1,2,3,4,5,6]
x2 = ['월','화','수','목','금','토','일']
y = [1950,2350,1850, 2200,3800,2800,2000]
y2 = [1750,2150,2550, 2300,2400,1900,1600]
```

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



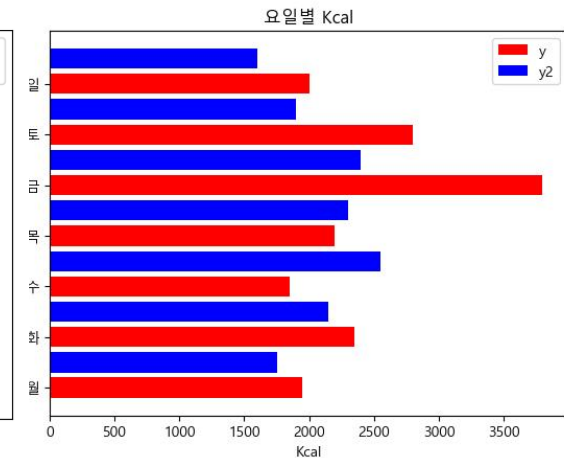
```
x = [0,2,4,6,8,10,12]
x1 = [1,3,5,7,9,11,13]
x2 = ['월','화','수','목','금','토','일']
y = [1950,2350,1850, 2200,3800,2800,2000]
y2 = [1750,2150,2550, 2300,2400,1900,1600]
```

```
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

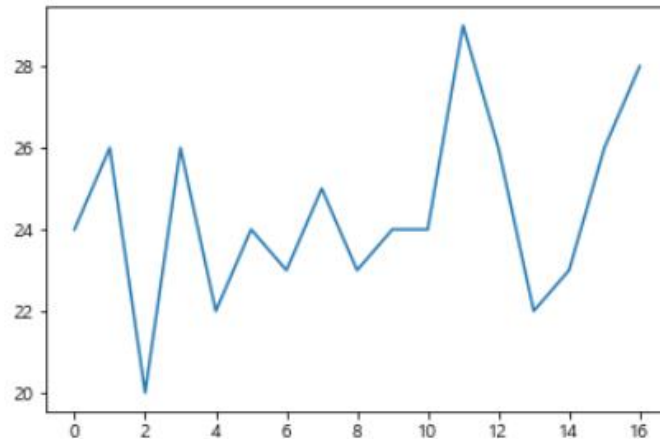
- 도별미세먼지.csv 파일을 읽어서 그래프를 그리시오.

```
#기본 그래프
import matplotlib.pyplot as plt

x = list(range(len(dic)))
y = list(dic.values())

print('\n')
print('기본 그래프')
plt.plot(y)
plt.show()
```

기본 그래프



```
#한글 폰트 사용
from matplotlib import font_manager, rc
font_name =
font_manager.FontProperties(fname="c:/Windows/Font
s/malgun.ttf").get_name()
rc('font', family=font_name)
```

```
#기본 그래프 - x축 표시

print('기본 그래프2')
plt.plot(y)
plt.xticks(x, list(dic.keys()))
plt.show()
```

기본 그래프2

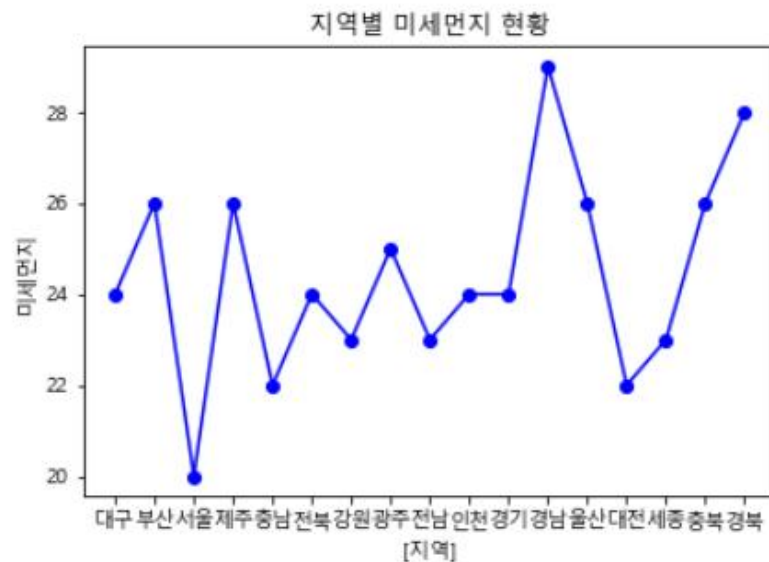


해결문제

- 도별미세먼지.csv 파일을 읽어서 그래프를 그리시오.

```
#기본 그래프 - 제목  
print('기본 그래프3')  
plt.plot(y, 'bo-')  
plt.xticks(x, list(dic.keys()))  
  
plt.title('지역별 미세먼지 현황')  
plt.xlabel('[지역]')  
plt.ylabel('미세먼지')  
plt.show()
```

기본 그래프3



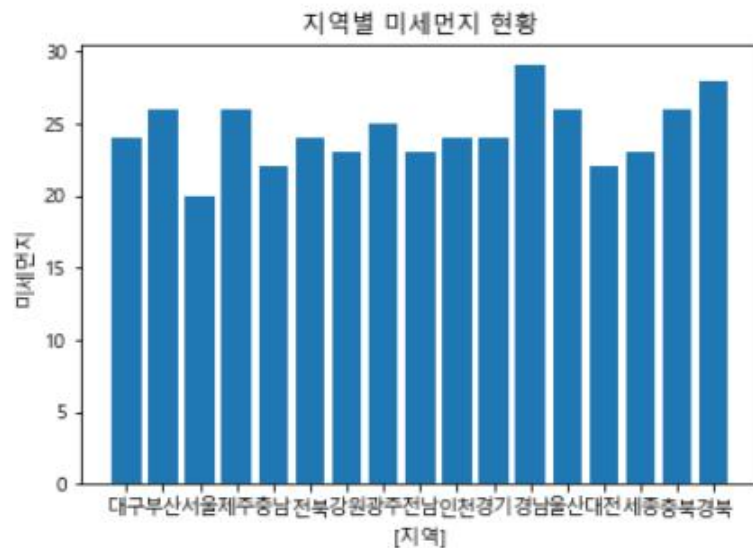
해결문제

- 도별미세먼지.csv 파일을 읽어서 그래프를 그리시오.

```
#기본 그래프 - 막대 그래프
print('기본 막대 그래프')
plt.bar(x,y)
plt.xticks(x, list(dic.keys()))

plt.title('지역별 미세먼지 현황')
plt.xlabel('[지역]')
plt.ylabel('미세먼지')
plt.show()
```

기본 막대 그래프



해결문제

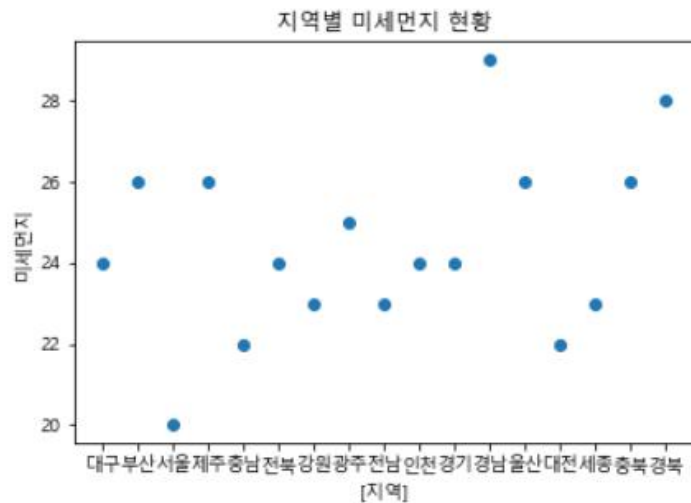
- 도별미세먼지.csv 파일을 읽어서 그래프를 그리시오.

```
#기본 그래프 - 산점도 그래프
print('기본 막대 그래프')
plt.scatter(x,y)
plt.xticks(x, list(dic.keys()))

plt.title('지역별 미세먼지 현황')
plt.xlabel('[지역]')
plt.ylabel('미세먼지')

#plt.ylim([0, 30])
plt.show()
```

기본 막대 그래프

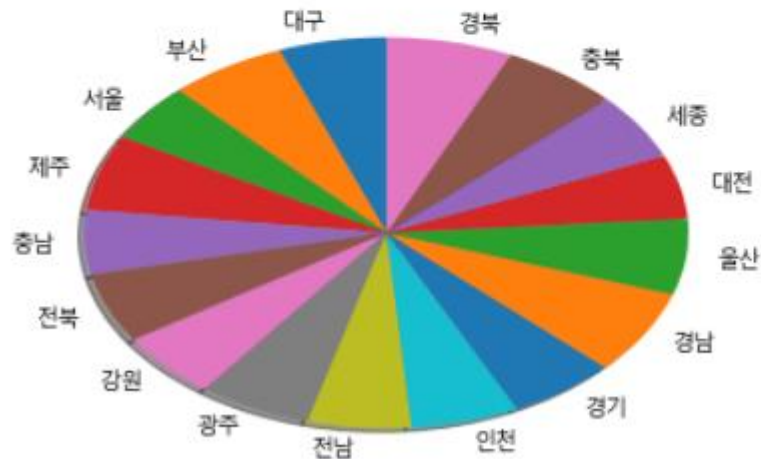


해결문제

- 도별미세먼지.csv 파일을 읽어서 그래프를 그리시오.

```
#기본 그래프 - 원형 그래프  
print('원형 그래프')
```

```
plt.pie(y, labels=list(dic.keys()), shadow=True, startangle=90)  
plt.show()
```



해결문제

- 부산시기온.csv자료를 읽어서 다음과 같이 시각화 하시오.

