

▼ TABA 인공지능 이론강의

matplotlib

import

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

한국어 사용하기

```
# 로컬에서
plt.rcParams['font.family'] = 'Malgun Gothic'
plt.rcParams['axes.unicode_minus'] = False
```

```
# 코랩에서
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
```

```
☞ Reading package lists... Done
Building dependency tree
Reading state information... Done
fonts-nanum is already the newest version (20180306-3).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
/usr/share/fonts: caching, new cache contents: 0 fonts, 1 dirs
/usr/share/fonts/truetype: caching, new cache contents: 0 fonts, 3 dirs
/usr/share/fonts/truetype/humor-sans: caching, new cache contents: 1 fonts, 0 dirs
/usr/share/fonts/truetype/liberation: caching, new cache contents: 16 fonts, 0 dirs
/usr/share/fonts/truetype/nanum: caching, new cache contents: 10 fonts, 0 dirs
/usr/local/share/fonts: caching, new cache contents: 0 fonts, 0 dirs
/root/.local/share/fonts: skipping, no such directory
/root/.fonts: skipping, no such directory
/usr/share/fonts/truetype: skipping, looped directory detected
/usr/share/fonts/truetype/humor-sans: skipping, looped directory detected
/usr/share/fonts/truetype/liberation: skipping, looped directory detected
/usr/share/fonts/truetype/nanum: skipping, looped directory detected
/var/cache/fontconfig: cleaning cache directory
/root/.cache/fontconfig: not cleaning non-existent cache directory
/root/.fontconfig: not cleaning non-existent cache directory
fc-cache: succeeded
```

런타임 다시시작

```
plt.rc('font', family='NanumBarunGothic')
```

우리가 배우게 될 것들

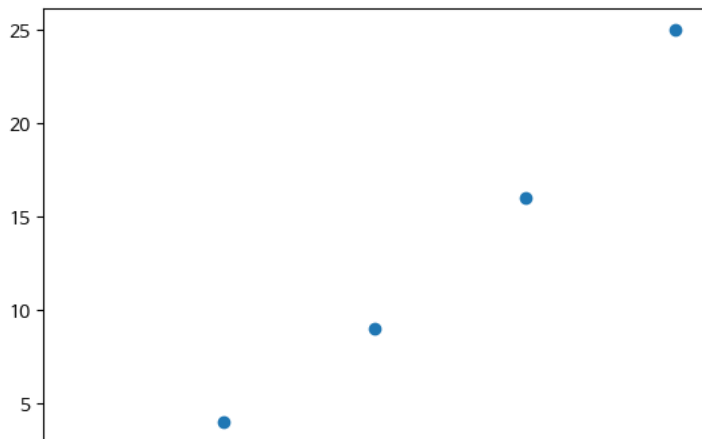
1. Scatter
2. Plot
3. Bar
4. Histogram
5. Imshow

▼ scatter

x랑 y직접 만들기

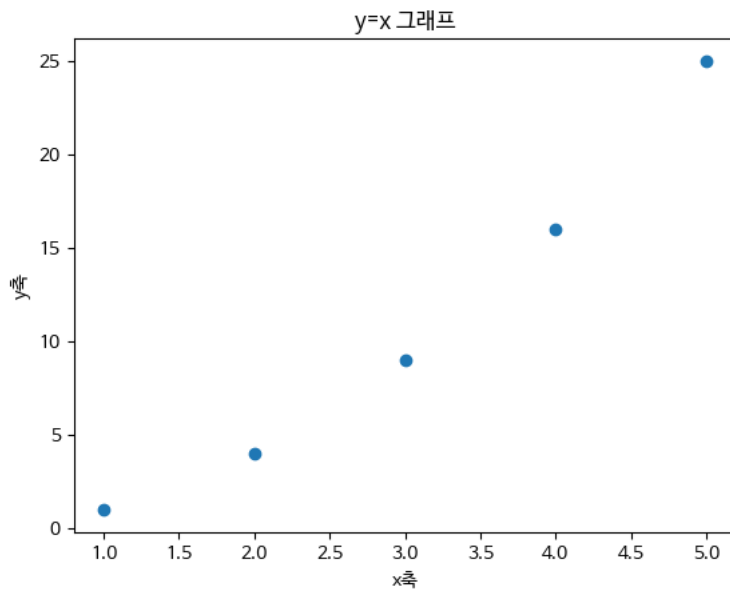
```
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]
```

```
plt.scatter(x,y)
plt.show()
```

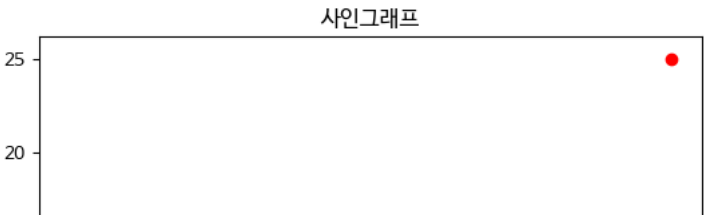


그래프 색 변환 / 제목쓰기 / 레이블 설정

```
# 색 변환
plt.scatter(x,y)
# 제목 표현
plt.title('y=x 그래프')
# x축 레이블 설정
plt.xlabel('x축')
# y축 레이블 설정
plt.ylabel('y축')
plt.show()
```



```
# 색 변환
plt.scatter(x,y, color = 'red')
# 제목 표현
plt.title('사인그래프')
# x축 레이블 설정
plt.xlabel('x축')
# y축 레이블 설정
plt.ylabel('y축')
plt.show()
```

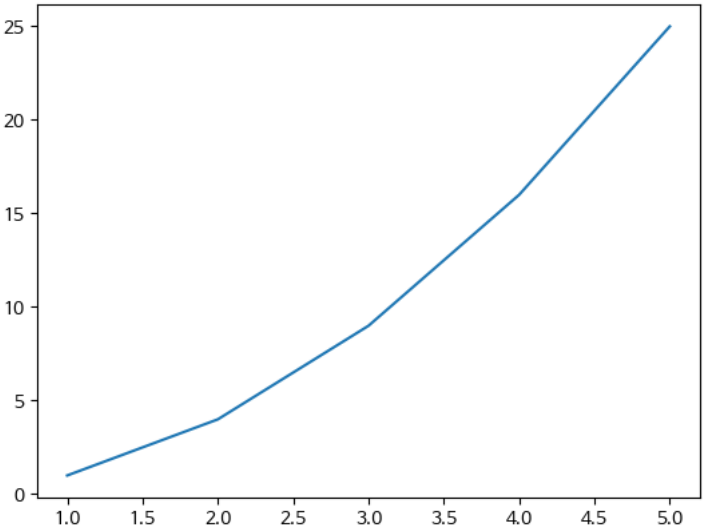


plot

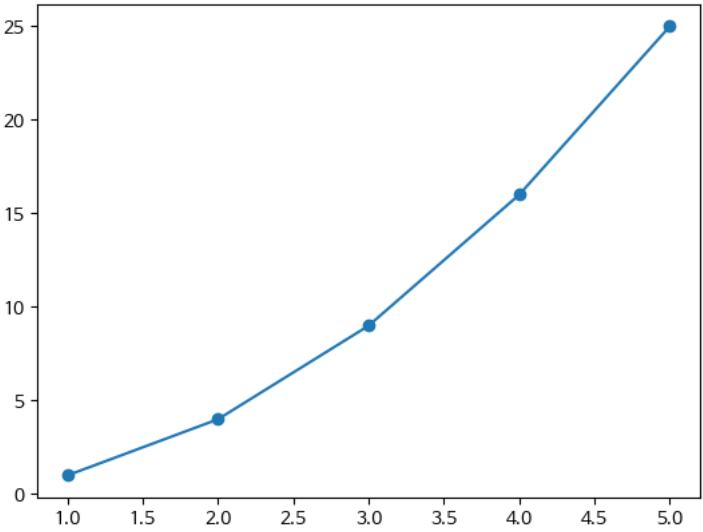
x랑 y를 직접 설정해서 만들기

```
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]
```

```
plt.plot(x,y)
plt.show()
```



```
plt.plot(x,y)
plt.scatter(x,y)
plt.show()
```



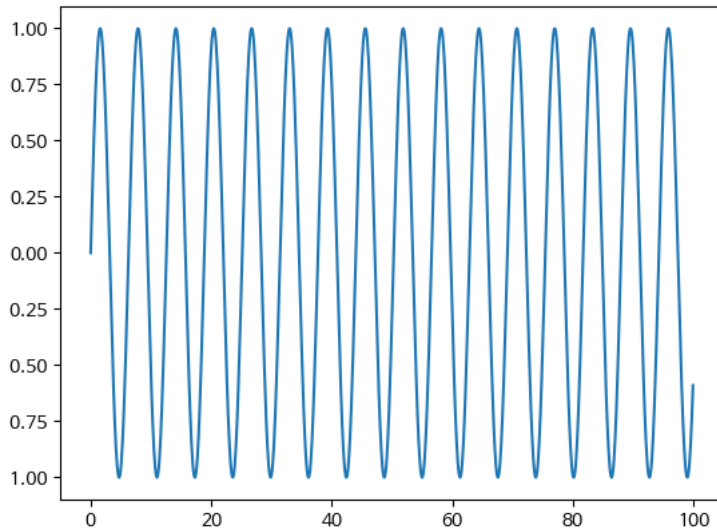
데이터 만들기 (sin)

```
x = np.arange(0, 100, 0.1)
y = np.sin(x)
```

간단하게 plot해서 표현하기

```
plt.plot(x,y)
plt.show()
```

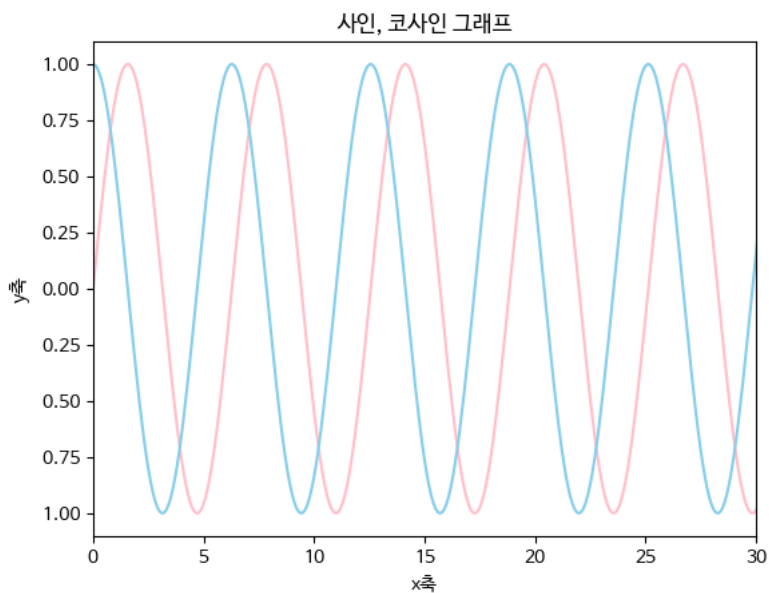
```
/usr/local/lib/python3.9/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
fig.canvas.print_figure(bytes_io, **kw)
```



Cosine 그래프 추가

```
x = np.arange(0, 100, 0.1)
sin_y = np.sin(x)
cos_y = np.cos(x)
```

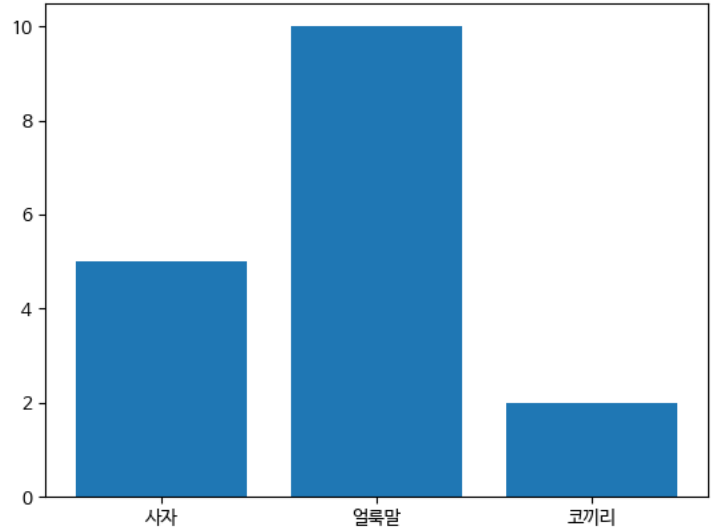
```
# 색 변환
plt.plot(x, sin_y, color = 'pink')
plt.plot(x, cos_y, color = 'skyblue')
# 제목 표현
plt.title('사인, 코사인 그래프')
# x축 레이블 설정
plt.xlabel('x축')
# y축 레이블 설정
plt.ylabel('y축')
# x 좌표 제한하기
plt.xlim([0,30])
plt.show()
```



▼ bar

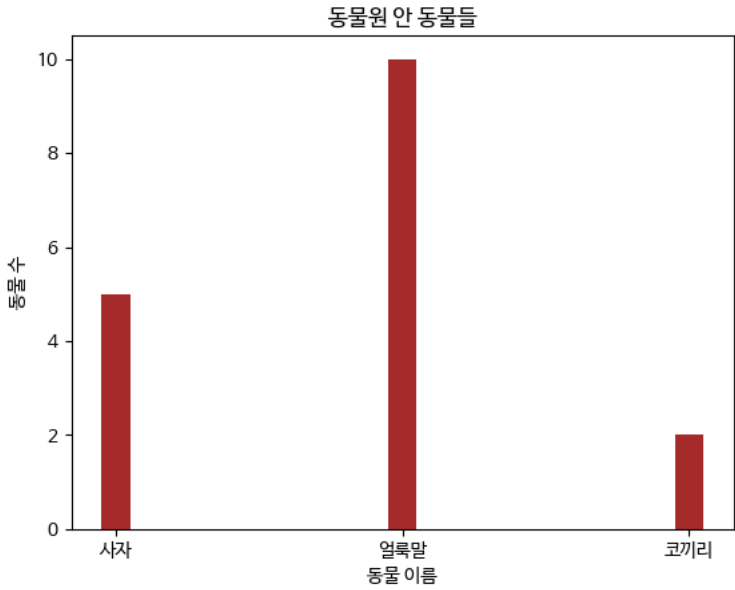
```
animal = ['사자', '얼룩말', '코끼리']
no_animal = [5, 10, 2]
```

```
plt.bar(animal,no_animal)
plt.show()
```



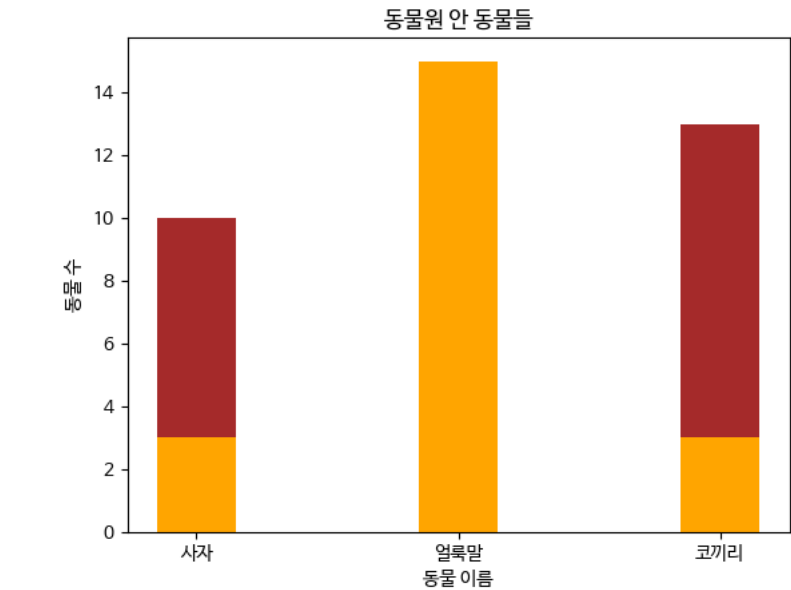
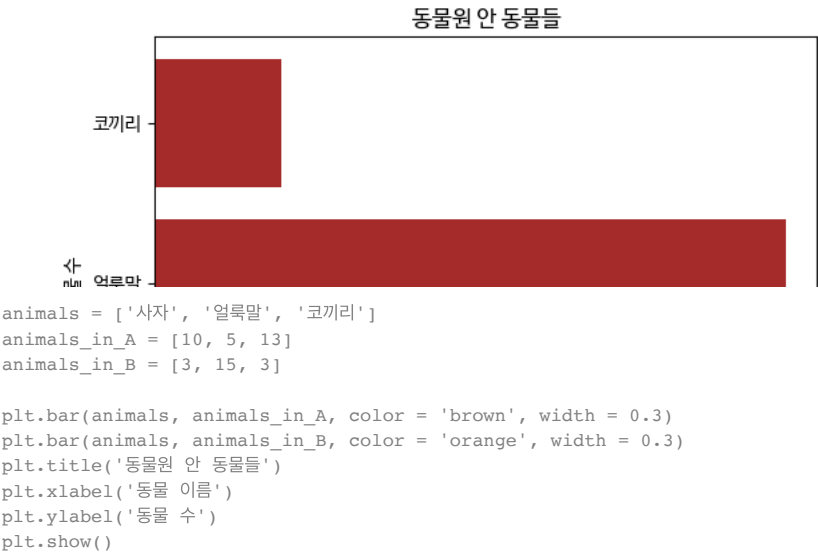
그래프를 꾸며줍니다

```
plt.bar(animal,no_animal, color = 'brown', width = 0.1)
plt.title('동물원 안 동물들')
plt.xlabel('동물 이름')
plt.ylabel('동물 수')
plt.show()
```



세로형 막대그래프

```
plt.barh(animal,no_animal, color = 'brown')
plt.title('동물원 안 동물들')
plt.xlabel('동물 이름')
plt.ylabel('동물 수')
plt.show()
```



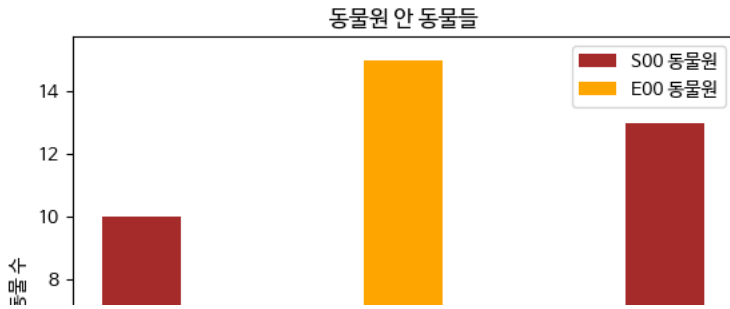
문제점이 존재

- 1. 각 색깔의 그래프가 뭘 나타내는지 모르겠음
- 2. 일단 B의 그래프가 A 위에 그려져서 가림

일단 각 색깔별로 뭘 나타내는지 나타내보자

```
animals = ['사자', '얼룩말', '코끼리']
animals_in_A = [10, 5, 13]
animals_in_B = [3, 15, 3]

plt.bar(animals, animals_in_A, color = 'brown', width = 0.3, label = 'S00 동물원')
plt.bar(animals, animals_in_B, color = 'orange', width = 0.3, label = 'E00 동물원')
plt.title('동물원 안 동물들')
plt.xlabel('동물 이름')
plt.ylabel('동물 수')
plt.legend(loc='best')
plt.show()
```



B 그래프를 옆으로 치워보자

그전에 일단 이렇게 x좌표를 만들어 주는게 필요하다

```

4 1 1
4 1 1
4 1 1

x = np.arange(1,4,1)
print(x)
w = 0.3
print(x+w)

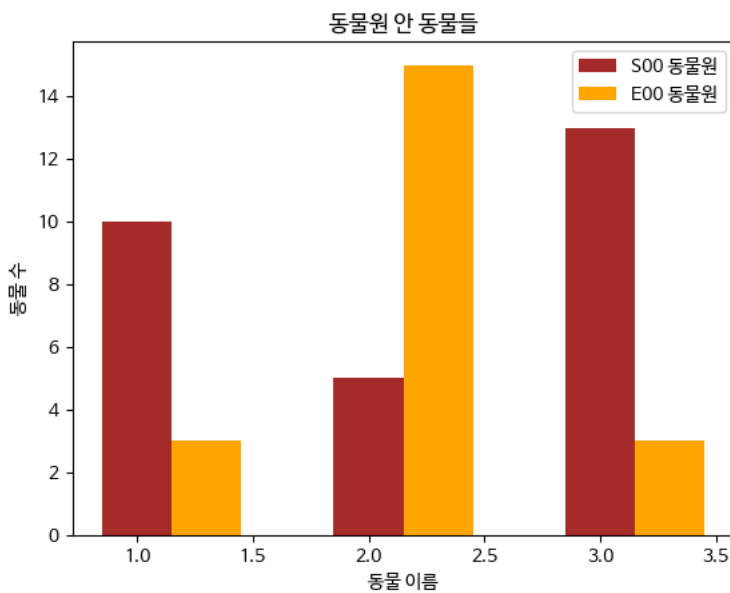
[1 2 3]
[1.3 2.3 3.3]

animals = ['사자', '얼룩말', '코끼리']
animals_in_A = [10, 5, 13]
animals_in_B = [3, 15, 3]

w = 0.3

plt.bar([1,2,3], animals_in_A, color = 'brown', width = w, label = 'S00 동물원')
plt.bar([1+w,2+w,3+w], animals_in_B, color = 'orange', width = w, label = 'E00 동물원')
plt.title('동물원 안 동물들')
plt.xlabel('동물 이름')
plt.ylabel('동물 수')
plt.legend(loc='best')
plt.show()

```



제대로 나왔으니깐 이번에는 x 좌표에 저 보기 싫은 숫자를 우리가 원하는 것들로 바꿔보자

```

animals = ['사자', '얼룩말', '코끼리']
animals_in_A = [10, 5, 13]
animals_in_B = [3, 15, 3]

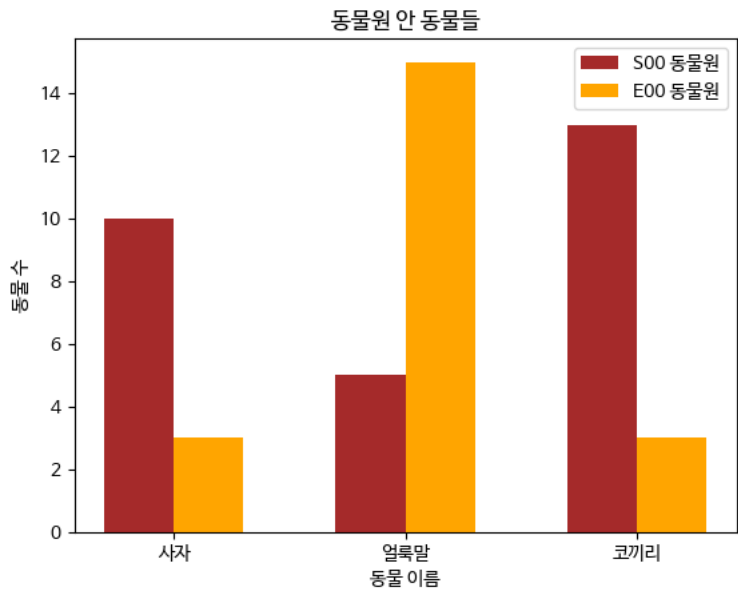
x = np.arange(1,4,1)
w = 0.3

plt.bar(x, animals_in_A, color = 'brown', width = w, label = 'S00 동물원')
plt.bar(x+w, animals_in_B, color = 'orange', width = w, label = 'E00 동물원')
plt.xticks(x+w/2, animals)

plt.title('동물원 안 동물들')
plt.xlabel('동물 이름')
plt.ylabel('동물 수')

```

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



동일하게 y 좌표도 바꿔줄 수 있겠다

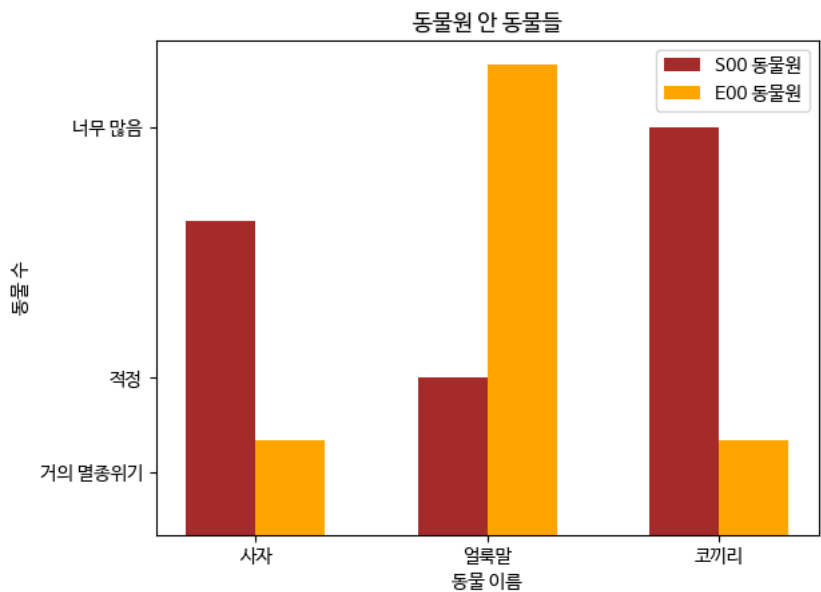
```
animals = ['사자', '얼룩말', '코끼리']
animals_in_A = [10, 5, 13]
animals_in_B = [3, 15, 3]

x = np.arange(1,4,1)
y = ['거의 멸종위기', '적정', '너무 많음']
w = 0.3

plt.bar(x, animals_in_A, color = 'brown', width = w, label = 'S00 동물원')
plt.bar(x+w, animals_in_B, color = 'orange', width = w, label = 'E00 동물원')

plt.xticks(x+w/2, animals)
plt.yticks([2, 5, 13], y)

plt.title('동물원 안 동물들')
plt.xlabel('동물 이름')
plt.ylabel('동물 수')
plt.legend(loc='best')
plt.show()
```



▼ histogram

어디서 데이터 받아오기가 거시기하니까 그냥 데이터들을 만들자

일반적인 히스토그램 그래프 생성

A bar chart with the x-axis labeled 'number' ranging from 1.0 to 5.0 in increments of 0.5. The y-axis represents the frequency of observations, with major ticks at 0, 2, 4, 6, and 8. The bars show the following frequencies: 1.0 has 3 observations, 2.0 has 3 observations, 3.0 has 8 observations, 4.0 has 3 observations, and 5.0 has 9 observations.

| number | frequency |
|--------|-----------|
| 1.0 | 3 |
| 2.0 | 3 |
| 3.0 | 8 |
| 4.0 | 3 |
| 5.0 | 9 |

| Age Group | Number of People |
|-----------|------------------|
| 1.0 | 3 |
| 2.0 | 3 |
| 3.0 | 8 |
| 4.0 | 3 |
| 5.0 | 9 |

datad에 들어가 있는 원소들 각각의 개수를 찾아보자

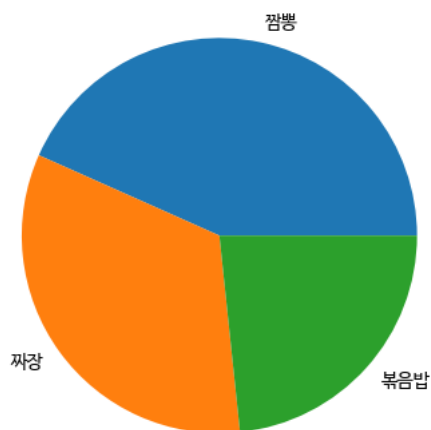
9/17

```
plt.pie(ds.values)
plt.show()
```



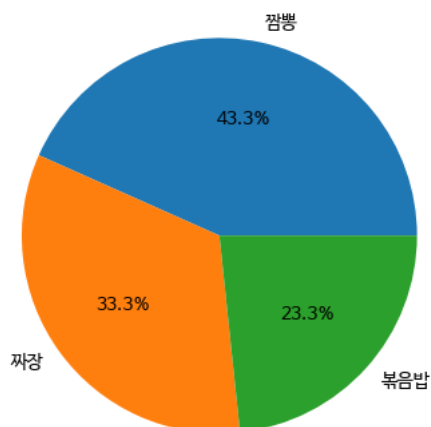
인덱스가 없어서 뭐가 얼마나 나타내는지 모르겠으니 그걸 표시해보자

```
plt.pie(ds.values, labels=ds.index)
plt.show()
```



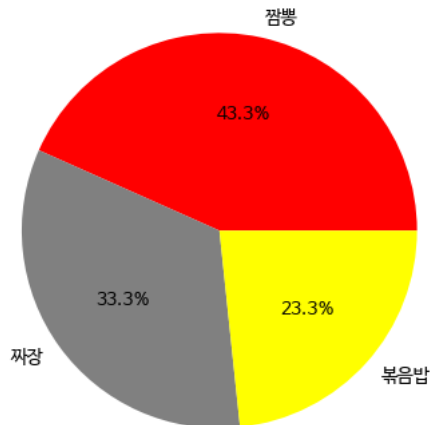
퍼센테이지로 표시하자

```
# autopct로 퍼센테이지 표현. 저기 숫자는 소수점 아래 몇번째 자리 숫자까지 표시할건지 알 수 있음
plt.pie(ds.values, labels=ds.index, autopct = '%.1f%%')
plt.show()
```



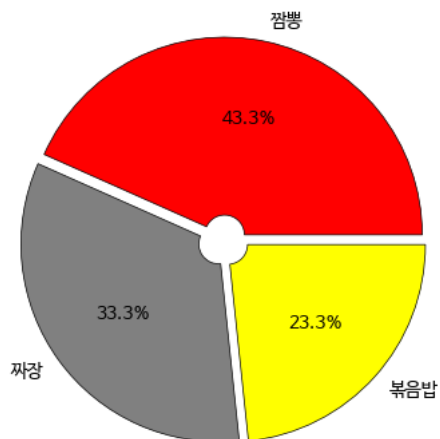
짜장은 짜장 색깔에, 짬뽕은 짬뽕 색깔에, 볶음밥은 볶음밥 색깔이 나와야 하지 않겠어?

```
colors = ['red', 'gray', 'yellow']
plt.pie(ds.values, labels=ds.index, autopct = '%.1f%%', colors=colors)
plt.show()
```



그래프 꾸미기(이건 옵션)

```
colors = ['red', 'gray', 'yellow']
plt.pie(ds.values, labels=ds.index,
        autopct = '%.1f%%',
        colors=colors,
        explode = [0.03, 0.03, 0.03],
        wedgeprops = {'width':0.9, 'edgecolor':'black', 'linewidth':0.5})
# 설명
# explode는 각 파이별로 얼마나 띄어지는지
# wedgeprops
# width: 1일때 완벽한 구, 0.9일때 안쪽 10%가 빈칸인 도넛
# edgecolor: 외곽선 색
# linewidth: 외곽선 두께
plt.show()
```



▼ 이미지 표시

```
import cv2
img = cv2.imread('img.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

<matplotlib.image.AxesImage at 0x7fd7d1732250>



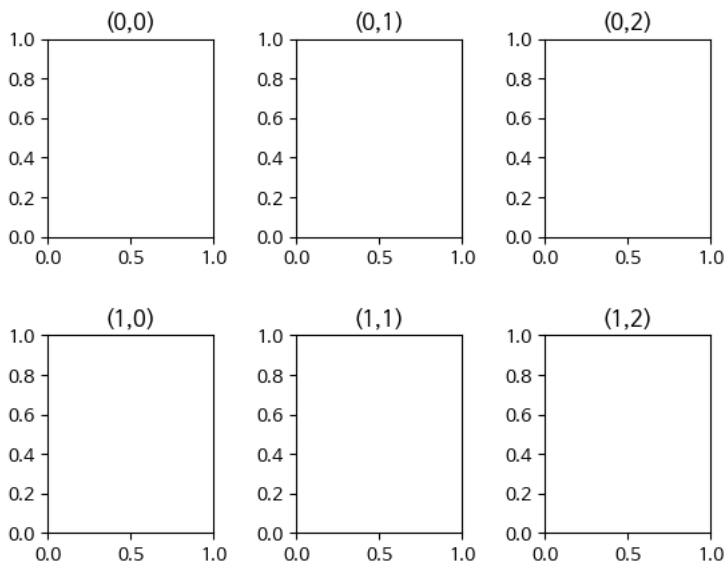
▼ 여러개 그래프 표시



여러 그래프를 한번에 그려서 지금까지 그린 그래프를 집대성해보자

일단 빈 강통 그래프를 그림

```
fig, ax = plt.subplots(2,3)
# 이름설정
ax[0,0].set_title('(0,0)')
ax[0,1].set_title('(0,1)')
ax[0,2].set_title('(0,2)')
ax[1,0].set_title('(1,0)')
ax[1,1].set_title('(1,1)')
ax[1,2].set_title('(1,2)')
# sub그래프간 간격설정
plt.subplots_adjust(wspace=0.5, hspace=0.5)
plt.show()
```



```
fig, ax = plt.subplots(2,3)
```

```
# 0,0 scatter
scat_x = [1, 2, 3, 4, 5]
scat_y = [1, 4, 9, 16, 25]
# 색 변환
ax[0,0].scatter(scat_x,scat_y,s=1)
# 제목 표현
ax[0,0].set_title('사인그래프')
# x축 레이블 설정
ax[0,0].set_xlabel('x축')
# y축 레이블 설정
ax[0,0].set_ylabel('y축')
```

```
# 0,1 sin,cos plot
x = np.arange(0, 100, 0.1)
```

```
sin_y = np.sin(x)
cos_y = np.cos(x)
# 색 변환
ax[0,1].plot(x, sin_y, color = 'pink')
ax[0,1].plot(x, cos_y, color = 'skyblue')
# 제목 표현
ax[0,1].set_title('사인, 코사인 그래프')
# x축 레이블 설정
ax[0,1].set_xlabel('x축')
# y축 레이블 설정
ax[0,1].set_ylabel('y축')
# x 좌표 제한하기
ax[0,1].set_xlim([0,30])


# 0,2 동물원 동물 숫자
animals = ['사자', '얼룩말', '코끼리']
animals_in_A = [10, 5, 13]
animals_in_B = [3, 15, 3]


x = np.arange(1,4,1)
y = ['거의 멸종위기', '적정', '너무 많음']
w = 0.3


ax[0,2].bar(x, animals_in_A, color = 'brown', width = w, label = 'S00 동물원')
ax[0,2].bar(x+w, animals_in_B, color = 'orange', width = w, label = 'E00 동물원')


ax[0,2].set_xticks(x+w/2, animals)
ax[0,2].set_yticks([2, 5, 13], y)


ax[0,2].set_title('동물원 안 동물들')
ax[0,2].set_xlabel('동물 이름')
ax[0,2].set_ylabel('동물 수')
ax[0,2].legend(loc='best')


# 1,0 히스토그램
data = [1,1,1,2,2,2,3,3,3,3,3,3,3,3,4,4,4,5,5,5,5,5,5,5,5]
ax[1,0].hist(data, bins=5, width = 0.5, edgecolor='black')


# 1,1 짜장 짬뽕 파이
data = ['짜장', '짜장', '짜장', '짜장', '짜장', '짜장', '짜장', '짜장', '짜장', '짜장', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕', '짬뽕']
colors = ['red', 'gray', 'yellow']
ax[1,1].pie(ds.values, labels=ds.index,
            autopct = '%.1f%%',
            colors=colors,
            explode = [0.03, 0.03, 0.03],
            wedgeprops = {'width':0.9, 'edgecolor':'black', 'linewidth':0.5})

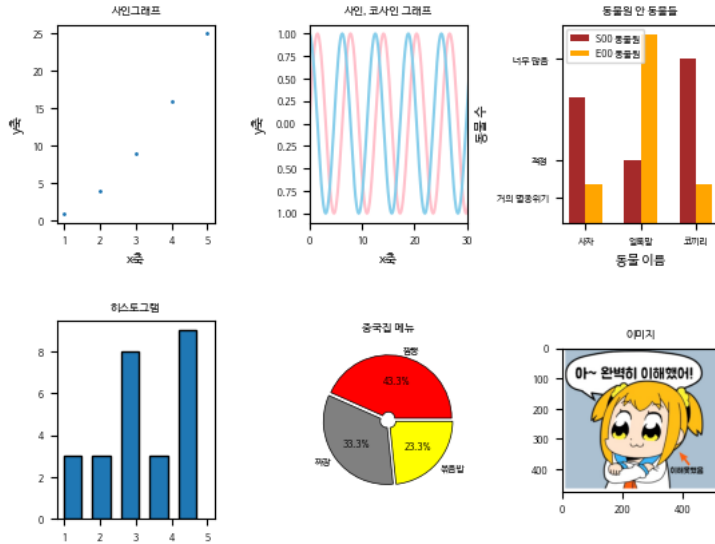

# 1,2 이미지 불러오기
img = cv2.imread('img.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
ax[1,2].imshow(img)


# 마지막 그래프 출력하기
plt.show()
```



```
# 1,2 이미지 불러오기
img = cv2.imread('img.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
ax[1,2].imshow(img)
ax[1,2].set_title('이미지')
```

```
# 마지막 그래프 출력하기
plt.show()
```



▼ 이미지 저장

```
fig, ax = plt.subplots(2,3)
plt.subplots_adjust(wspace=0.6, hspace=0.5)
plt.rc('font', size=5) # 기본 폰트 크기
plt.rc('axes', labelsz=7) # x,y축 label 폰트 크기
plt.rc('xtick', labelsz=5) # x축 눈금 폰트 크기
plt.rc('ytick', labelsz=5) # y축 눈금 폰트 크기
plt.rc('legend', fontsize=5) # 범례 폰트 크기
plt.rc('figure', titlesz=10) # figure title 폰트 크기
```

```
# 0,0 scatter
scat_x = [1, 2, 3, 4, 5]
scat_y = [1, 4, 9, 16, 25]
# 색 변환
ax[0,0].scatter(scat_x,scat_y,s=1)
# 제목 표현
ax[0,0].set_title('사인그래프')
# x축 레이블 설정
ax[0,0].set_xlabel('x축')
# y축 레이블 설정
ax[0,0].set_ylabel('y축')
```

```
# 0,1 sin,cos plot
x = np.arange(0, 100, 0.1)
sin_y = np.sin(x)
cos_y = np.cos(x)
# 색 변환
ax[0,1].plot(x, sin_y, color = 'pink')
ax[0,1].plot(x, cos_y, color = 'skyblue')
# 제목 표현
ax[0,1].set_title('사인, 코사인 그래프')
# x축 레이블 설정
ax[0,1].set_xlabel('x축')
# y축 레이블 설정
ax[0,1].set_ylabel('y축')
# x 좌표 제한하기
ax[0,1].set_xlim([0,30])
```

```
# 0,2 동물원 동물 숫자
animals = ['사자', '얼룩말', '코끼리']
animals_in_A = [10, 5, 13]
animals_in_B = [3, 15, 3]
```

```
<ipython-input-48-bc26ebaab074>:86: UserWarning: Glyph 8722 (\N{MINUS SIGN}) not
plt.savefig('plt_img.png')
```



