

파이썬을 이용한 시각화 기본

학습 목표

- matplotlib를 활용한 시각화에 대해 알아봅니다.

학습 내용

- 5. 타이틀과 레이블
- 6. Grid
- 7. 여러개 그래프 전체 타이틀 표시 - SuperTitle
- 8. Scatter Plot(산점도)
- 9. Alpha(투명도)
- 10. Barplot(막대그래프)
- 11. Horizontal Bars(수평 막대그래프)
- 12. 히스토그램
- 13. pie chart(원그래프)

목차

[05. 타이틀과 레이블](#)

[06. Grid](#)

[07. 여러개 그래프 전체 타이틀 표시 - SuperTitle](#)

[08. Scatter Plot\(산점도\)](#)

[09. Alpha\(투명도\)](#)

[10. Barplot\(막대그래프\)](#)

[11. Horizontal Bars\(수평 막대그래프\)](#)

[12. 히스토그램](#)

[13. pie chart\(원그래프\)](#)

In [11]:

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

print(matplotlib.__version__)
```

3.3.2

05. 타이틀과 레이블

[목차로](#)

In [12]:

```
import os, warnings
warnings.filterwarnings(action='ignore')
```

In [13]:

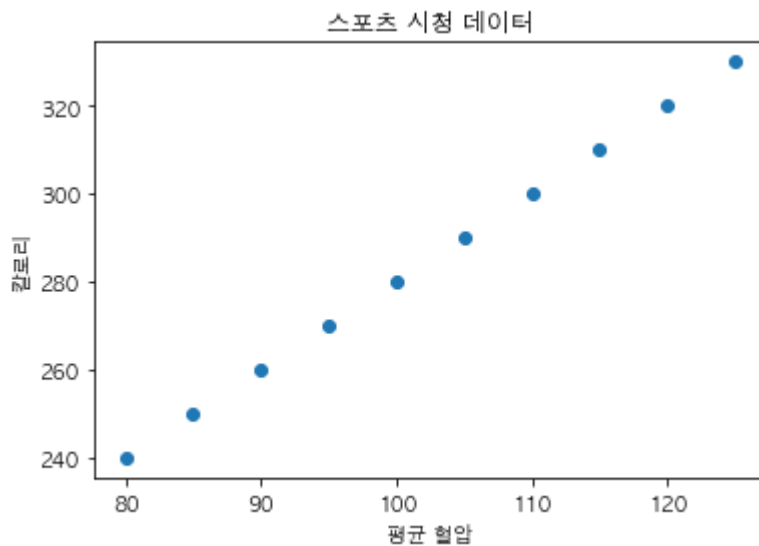
```
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.plot(x, y, 'o')

plt.title("스포츠 시청 데이터")
plt.xlabel("평균 혈압")
plt.ylabel("칼로리")
```

Out[13]:

Text(0, 0.5, '칼로리')



한글 표기

In [14]:

```
from matplotlib import font_manager, rc
import platform
import matplotlib.pyplot as plt
```

In [15]:

```

path = "C:/Windows/Fonts/malgun.ttf"
if platform.system() == "Windows":
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
elif platform.system()=="Darwin":
    rc('font', family='AppleGothic')
else:
    print("Unknown System")

matplotlib.rcParams['axes.unicode_minus'] = False

```

In [16]:

```

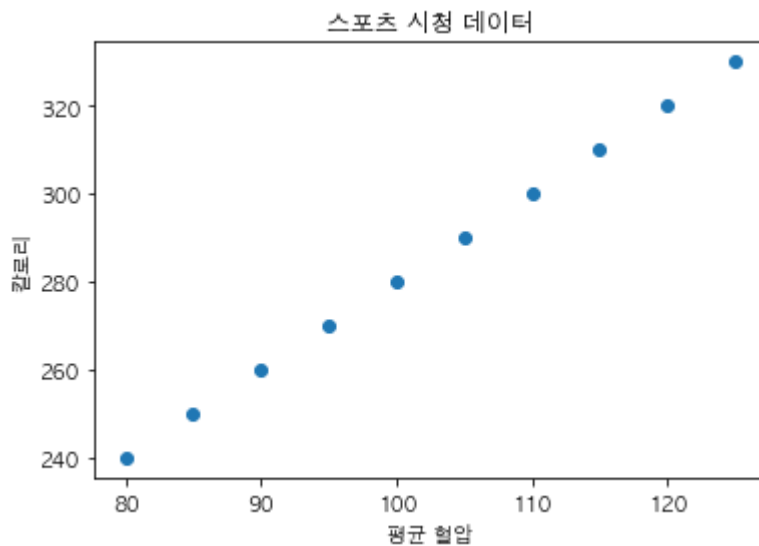
plt.plot(x, y, 'o')

plt.title("스포츠 시청 데이터")
plt.xlabel("평균 혈압")
plt.ylabel("칼로리")

```

Out[16]:

Text(0, 0.5, '칼로리')



06. Grid

- 격자, 모눈이라는 뜻이다. 내용을 구성하는 데 사용되는 일련의 교차하는 직선 또는 곡선으로 구성된 구조.

[목차로](#)

In [19]:

```
x = np.array([70, 75, 80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([220, 240, 260, 280, 300, 320, 340, 360, 380, 400, 420, 440])

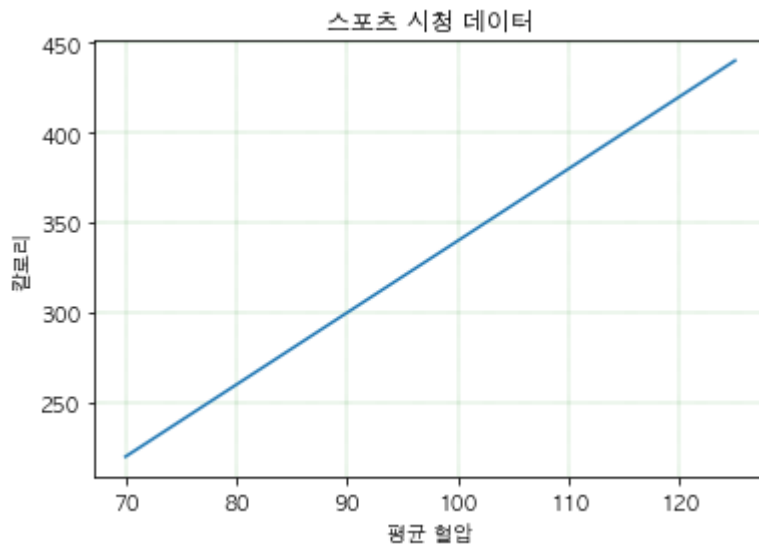
plt.title("스포츠 시청 데이터")
plt.xlabel("평균 혈압")
plt.ylabel("칼로리")

plt.grid(color = 'green', linestyle = '--', linewidth = 0.2)

plt.plot(x, y)
```

Out[19]:

[<matplotlib.lines.Line2D at 0x7fa255b533a0>]



07. 여러개 그래프 전체 타이틀 표시 - SuperTitle

목차로

- 전체 이미지의 상위 타이틀을 `subtitle()`를 이용하여 제목을 추가할 수 있다.

In [20]:

```

import matplotlib.pyplot as plt
import numpy as np

plt.figure(figsize=(8,5))

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

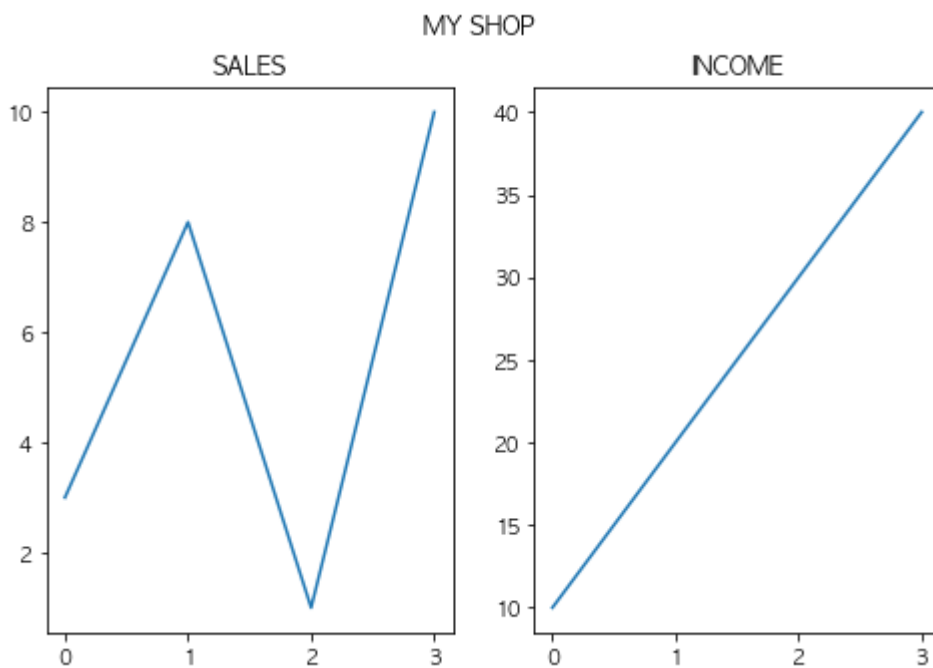
# 1행 2열, 첫번째
plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SALES")

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

# 1행 2열, 두번째
plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("INCOME")

plt.suptitle("MY SHOP")
plt.show()

```



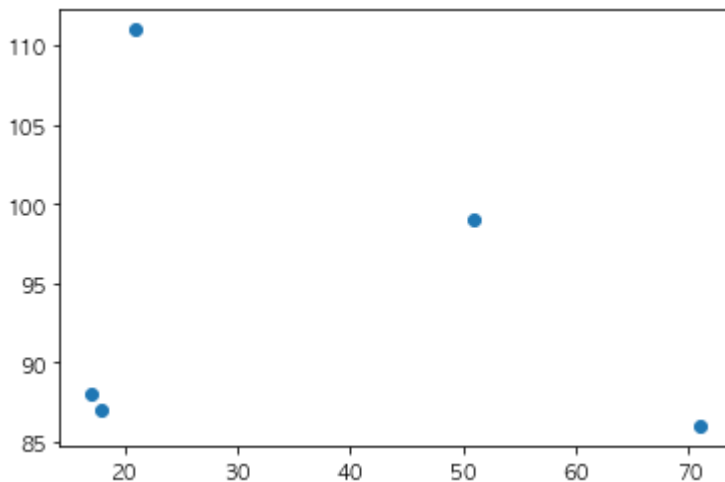
08. Scatter Plot(산점도)

- 직교 좌표계를 활용하여 좌표상의 점들을 표시.
- 두개 변수간의 관계를 나타낼 수 있다.

In [21]:

```
x = np.array([51,71,18,17,21])
y = np.array([99,86,87,88,111])

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



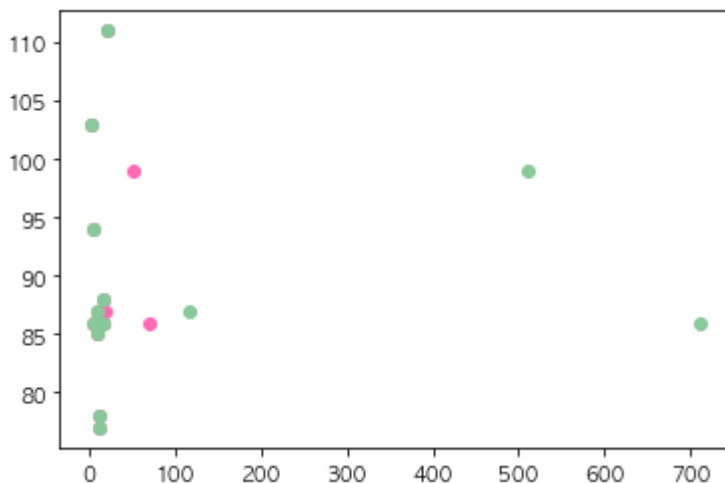
In [22]:

```
x = np.array([51,71,18,17,21,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y, color='hotpink')

x = np.array([511,711,118,17,21,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y, color='#88c999')
plt.show()
```



In [37]:

```

x = np.random.randint(10, size=10)
y = np.random.randint(80,100, size=10)
print(x, y)
print( len(x), len(y) )

# 색에 숫자로 맵핑
colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 100])

# 'viridis', 'plasma', 'inferno', 'magma', 'cividis'
plt.scatter(x, y, c=colors, cmap='magma', s=150)
plt.colorbar() # colormap 을 포함할 수 있다.

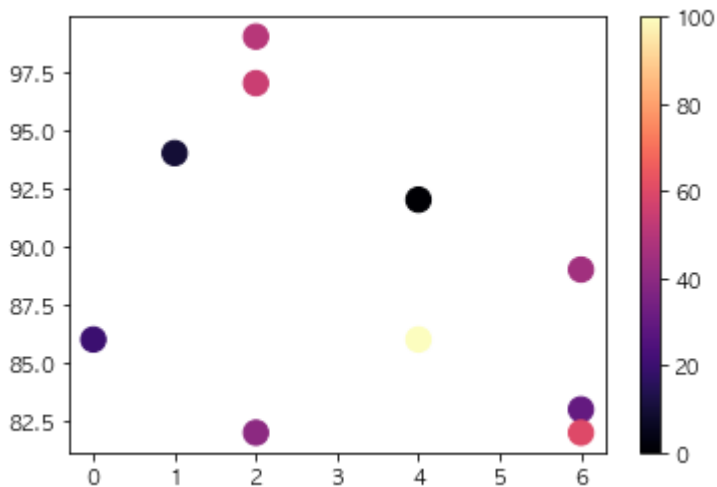
plt.show()

```

```

[4 1 0 6 2 6 2 2 6 4] [92 94 86 83 82 89 99 97 82 86]
10 10

```



사용가능한 colorMaps

- https://www.w3schools.com/python/matplotlib_scatter.asp
(https://www.w3schools.com/python/matplotlib_scatter.asp)

09. Alpha(투명도)

[목차로](#)

In [38]:

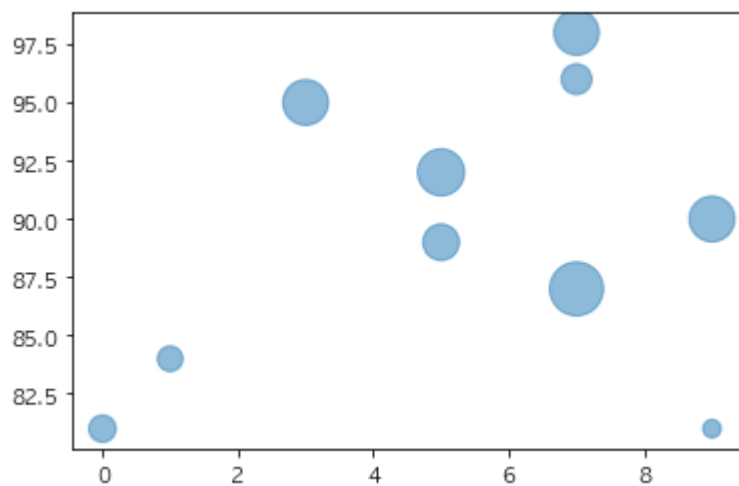
```
x = np.random.randint(10, size=10)
y = np.random.randint(80,100, size=10)

sizes = np.random.randint(20,800, size=10)
colors = np.random.randint(0,100, size=10)

plt.scatter(x, y, s=sizes, alpha=0.5)
```

Out[38]:

<matplotlib.collections.PathCollection at 0x7fa255fdc6a0>

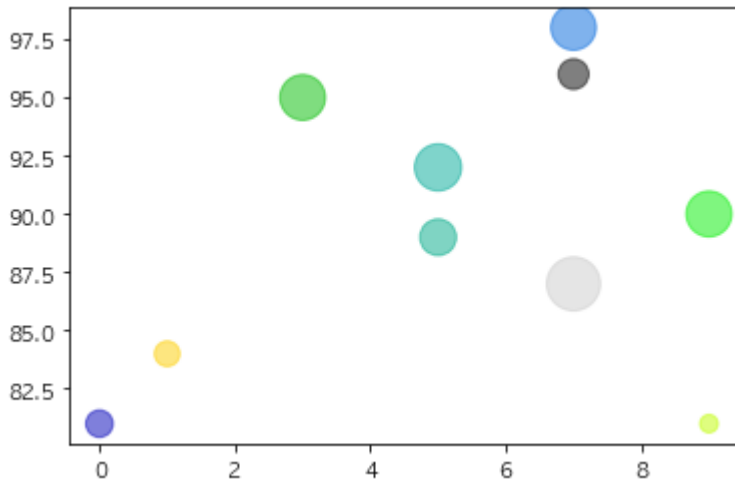


In [39]:

```
plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='nipy_spectral')
```

Out[39]:

```
<matplotlib.collections.PathCollection at 0x7fa2556186a0>
```



10. Barplot(막대그래프)

[목차로](#)

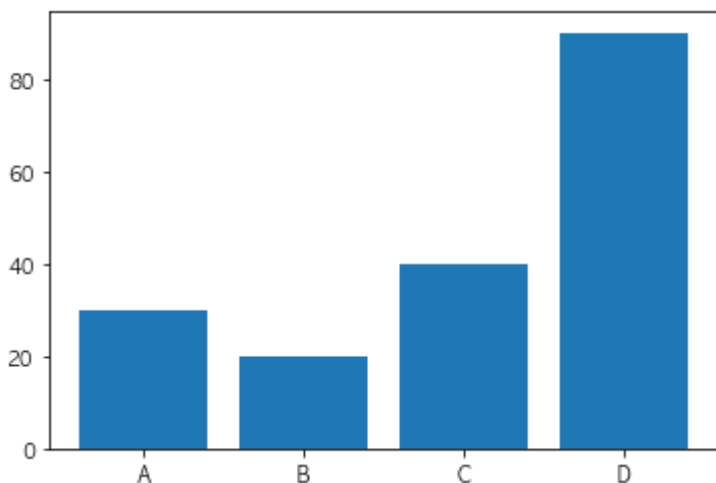
In [40]:

```
x = np.array(['A', 'B', 'C', 'D'])
y = np.array([30, 20, 40, 90])

plt.bar(x, y)
```

Out[40]:

```
<BarContainer object of 4 artists>
```



11. Horizontal Bars(수평 막대그래프)

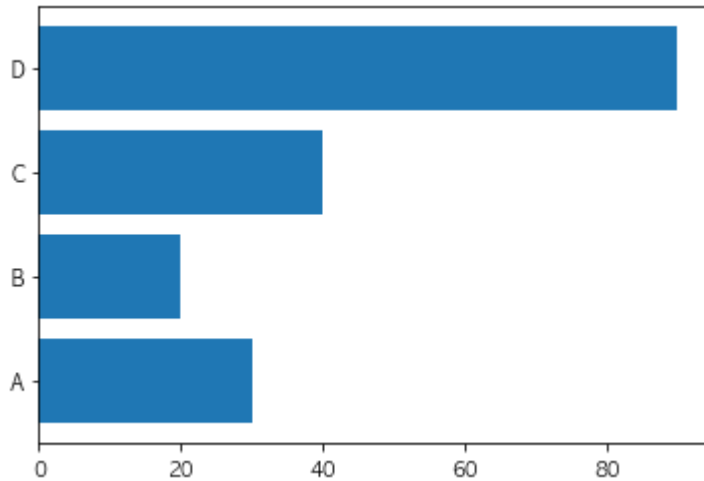
[목차로](#)

In [41]:

```
x = np.array(['A', 'B', 'C', 'D'])  
y = np.array([30, 20, 40, 90])  
  
plt.barh(x, y)
```

Out[41]:

<BarContainer object of 4 artists>

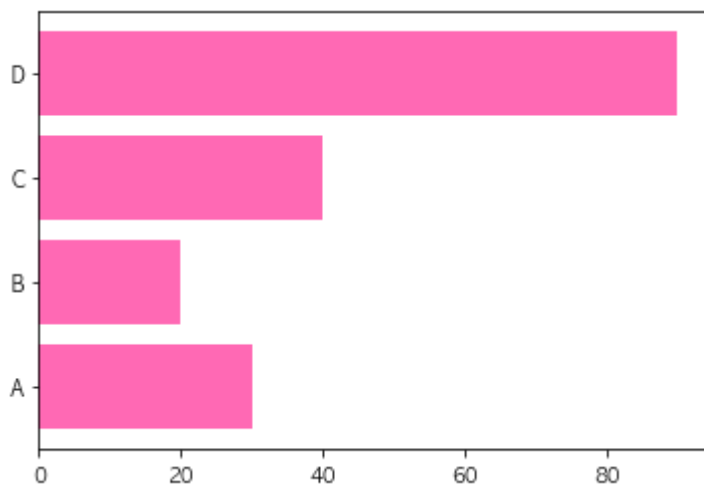


In [42]:

```
plt.barh(x, y, color = 'hotpink')
```

Out[42]:

<BarContainer object of 4 artists>

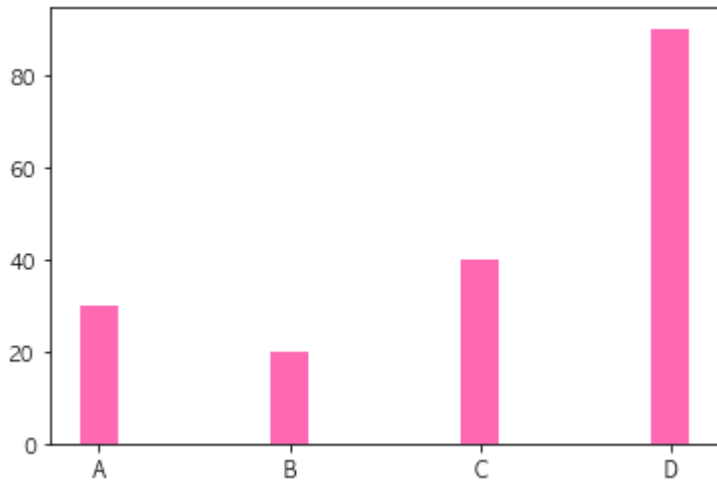


In [43]:

```
plt.bar(x, y, width=0.2, color = 'hotpink')
```

Out[43]:

<BarContainer object of 4 artists>

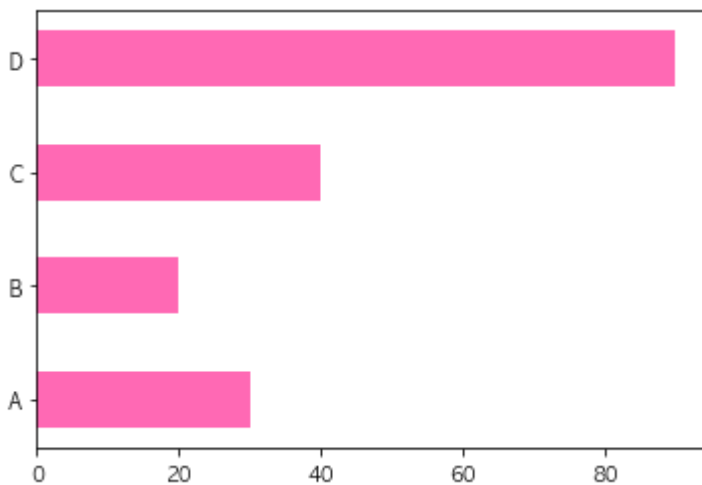


In [44]:

```
plt.barh(x, y, height=0.5, color = 'hotpink')
```

Out[44]:

<BarContainer object of 4 artists>



12. 히스토그램

- 연속형 값을 표시할 때 사용.
- 가로축이 계급, 세로축이 도수(구간의 값의 개수)를 의미

목차로

In [47]:

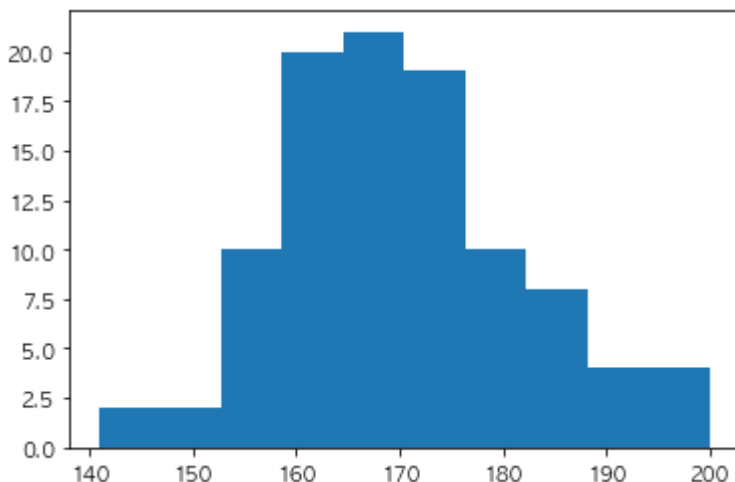
```
# 정규 분포를 따르는 값. 100개 생성
x = np.random.normal(loc=170, scale=10, size=100) # 평균, 표준편차, 개수

print(x)
plt.hist(x)
```

```
[156.12788779 178.90253703 167.17167432 183.52741494 188.28774817
171.79528532 154.3817283 172.54694927 178.3574289 163.31614622
189.40802549 169.82088829 147.21717751 175.96162781 172.19876431
141.0427876 181.6818787 165.78343486 161.10341166 171.2153579
174.56300008 172.8629086 153.47314016 164.76651659 163.12564621
166.22190475 195.80680962 179.34972119 161.55125637 185.53603227
160.17510509 171.9781842 155.7492202 173.07706453 182.70465225
173.03008426 199.88254193 186.59681074 167.29152933 192.69856814
166.9008666 152.74870256 172.63892627 198.7390959 178.72185355
166.91841734 165.256724 191.6775816 178.4572695 170.59499679
154.15740462 184.43898874 182.93683876 164.39422982 162.23823758
155.4453458 158.91456544 183.67377921 167.92093923 165.27823257
158.0027081 168.78619824 160.22539389 171.00214604 177.00295286
169.77931702 194.18370859 174.02945868 184.93137916 176.13749728
161.61562828 165.43173253 163.84807732 159.8808983 160.95643104
168.62728203 153.65721634 167.27051061 171.57053474 164.09782406
169.23427811 163.32363808 163.0657209 178.91662181 173.61615034
168.76290626 158.99050311 161.76076606 162.01808001 165.68582644
176.33344797 156.62513335 145.96588253 167.08600455 161.1053922
165.26767123 177.88888914 154.69757884 181.04147944 170.65501246]
```

Out[47]:

```
(array([ 2.,  2., 10., 20., 21., 19., 10.,  8.,  4.,  4.]),
array([141.0427876 , 146.92676303, 152.81073847, 158.6947139 ,
        164.57868933, 170.46266477, 176.3466402 , 182.23061563,
        188.11459107, 193.9985665 , 199.88254193]),
<BarContainer object of 10 artists>)
```



13 pie chart(원 그래프)

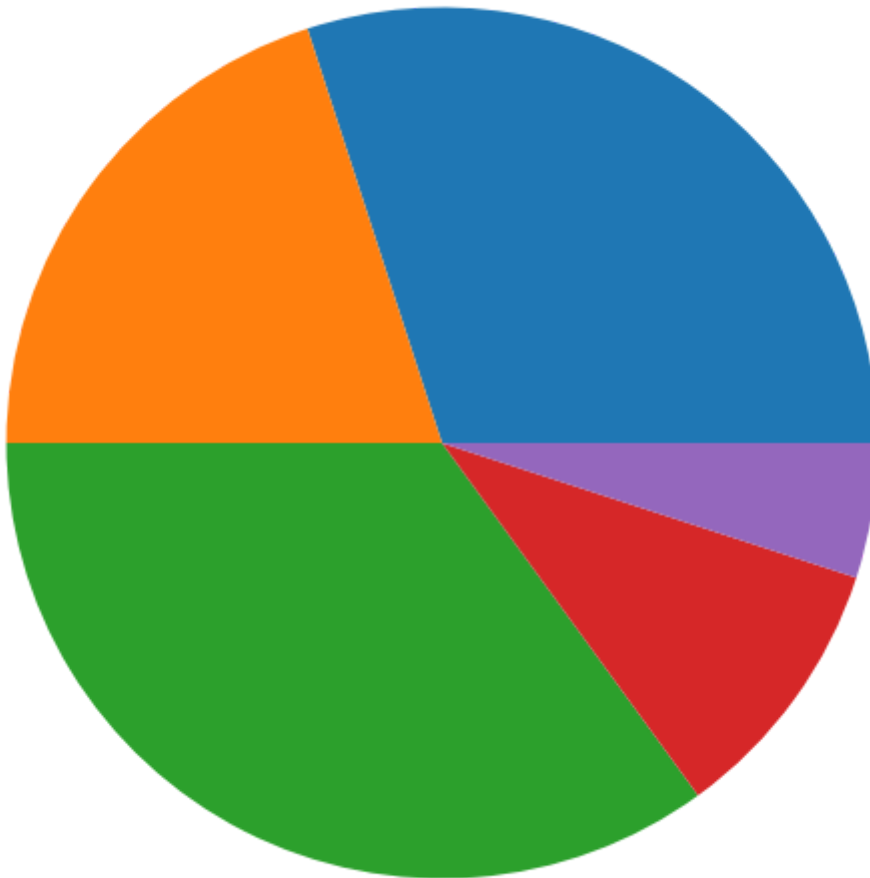
[matplotlib_B.ipynb](#)
[목차로](#)

In [48]:

```
y = np.array([30,20,35,10, 5])  
plt.figure(figsize=(10,10))  
plt.pie(y)
```

Out[48]:

```
([<matplotlib.patches.Wedge at 0x7fa2562b14c0>,  
<matplotlib.patches.Wedge at 0x7fa2562b1e50>,  
<matplotlib.patches.Wedge at 0x7fa2561fedf0>,  
<matplotlib.patches.Wedge at 0x7fa2561fe1c0>,  
<matplotlib.patches.Wedge at 0x7fa2561fe0a0>],  
[Text(0.6465637441936395, 0.8899187180267094, ''),  
Text(-0.8899187482945419, 0.6465637025335369, ''),  
Text(-0.49938947630209474, -0.9801072140121813, ''),  
Text(0.8899187331606258, -0.6465637233635886, ''),  
Text(1.0864571863351944, -0.1720778377961938, '')[0]])
```



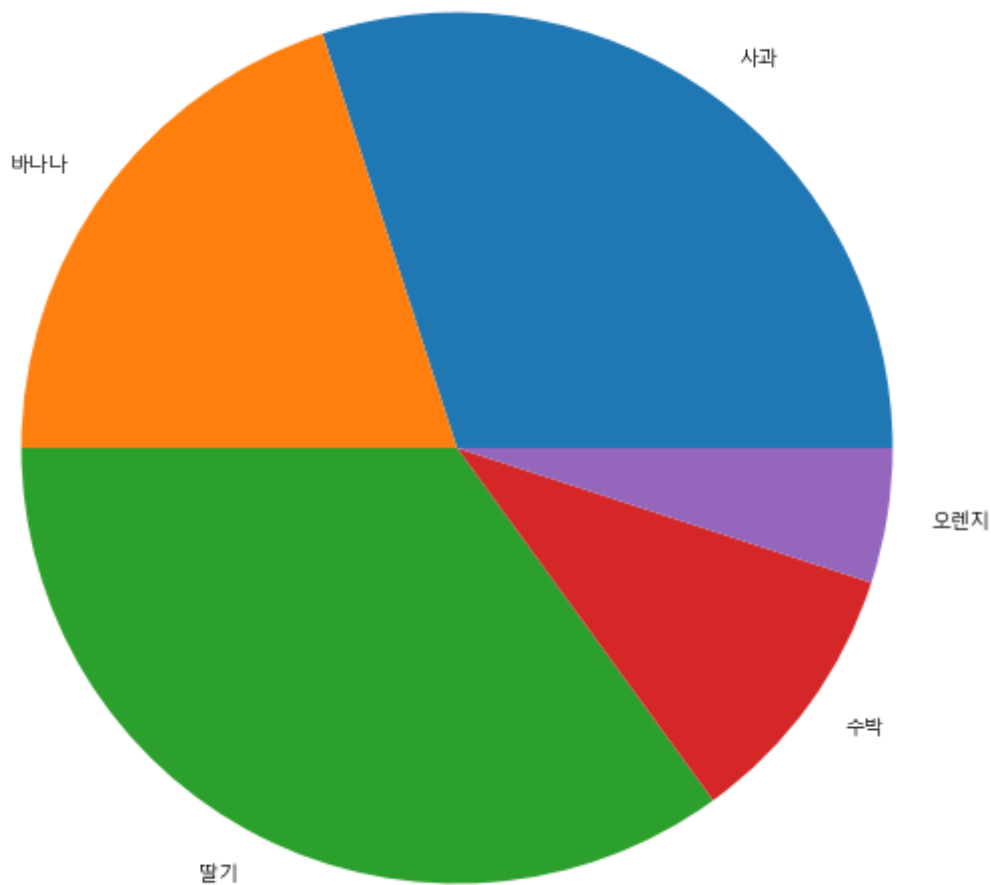
In [44]:

```
plt.figure(figsize=(10,10))

y = np.array([30,20,35,10, 5])
addlbl = ['사과', '바나나', '딸기', '수박', '오렌지']
plt.pie(y, labels=addlbl)
```

Out[44]:

```
([<matplotlib.patches.Wedge at 0x24f8dc17a30>,
<matplotlib.patches.Wedge at 0x24f8dc17e50>,
<matplotlib.patches.Wedge at 0x24f8dc25280>,
<matplotlib.patches.Wedge at 0x24f8dc256a0>,
<matplotlib.patches.Wedge at 0x24f8dc25b50>],
[Text(0.6465637441936395, 0.8899187180267095, '사과'),
Text(-0.8899187482945419, 0.6465637025335369, '바나나'),
Text(-0.49938947630209474, -0.9801072140121813, '딸기'),
Text(0.8899187331606258, -0.6465637233635886, '수박'),
Text(1.0864571863351944, -0.1720778377961938, '오렌지')])
```

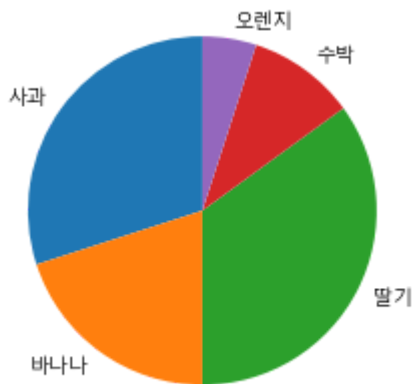


In [45]:

```
y = np.array([30,20,35,10, 5])  
addlbl = [ '사과', '바나나', '딸기', '수박', '오렌지' ]  
plt.pie(y, labels = addlbl, startangle = 90)
```

Out[45]:

```
([<matplotlib.patches.Wedge at 0x24f8ee39400>,  
 <matplotlib.patches.Wedge at 0x24f8ee398e0>,  
 <matplotlib.patches.Wedge at 0x24f8ee39d60>,  
 <matplotlib.patches.Wedge at 0x24f8ee461c0>,  
 <matplotlib.patches.Wedge at 0x24f8ee46640>],  
 [Text(-0.8899187180267095, 0.6465637441936395, '사과'),  
 Text(-0.6465637025335373, -0.8899187482945414, '바나나'),  
 Text(0.9801072140121813, -0.4993894763020948, '딸기'),  
 Text(0.6465637233635887, 0.8899187331606258, '수박'),  
 Text(0.17207783779619384, 1.0864571863351942, '오렌지')])
```



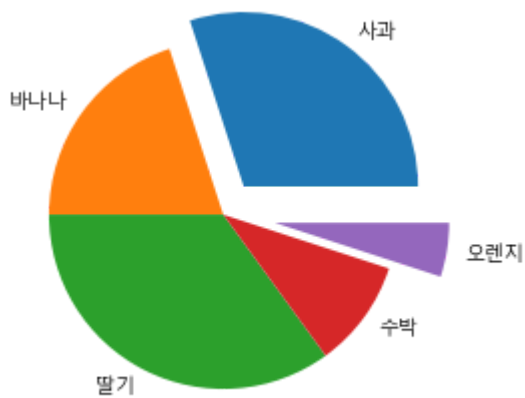
In [46]:

```
y = np.array([30,20,35,10, 5])
addlbl = [ '사과', '바나나', '딸기', '수박', '오렌지' ]
myexplode = [0.2, 0, 0, 0, 0.3]

plt.pie(y, labels = addlbl, explode = myexplode)
```

Out[46]:

```
([<matplotlib.patches.Wedge at 0x24f8ee8b340>,
 <matplotlib.patches.Wedge at 0x24f8ee8b760>,
 <matplotlib.patches.Wedge at 0x24f8ee8bb20>,
 <matplotlib.patches.Wedge at 0x24f8ee8bfa0>,
 <matplotlib.patches.Wedge at 0x24f8ee97490>],
 [Text(0.764120788592483, 1.051722121304293, '사과'),
 Text(-0.8899187482945419, 0.6465637025335369, '바나나'),
 Text(-0.49938947630209474, -0.9801072140121813, '딸기'),
 Text(0.8899187331606258, -0.6465637233635886, '수박'),
 Text(1.3827636916993382, -0.21900815719515573, '오렌지')])
```



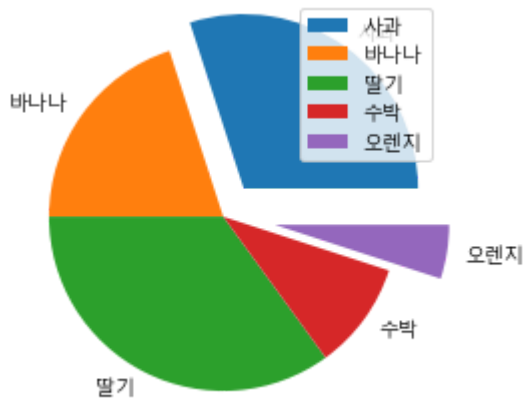
In [47]:

```
y = np.array([30,20,35,10, 5])
addlbl = [ '사과', '바나나', '딸기', '수박', '오렌지' ]
myexplode = [0.2, 0, 0, 0, 0.3]

plt.pie(y, labels = addlbl, explode = myexplode)
plt.legend()
```

Out[47]:

<matplotlib.legend.Legend at 0x24f8ee598e0>



In [48]:

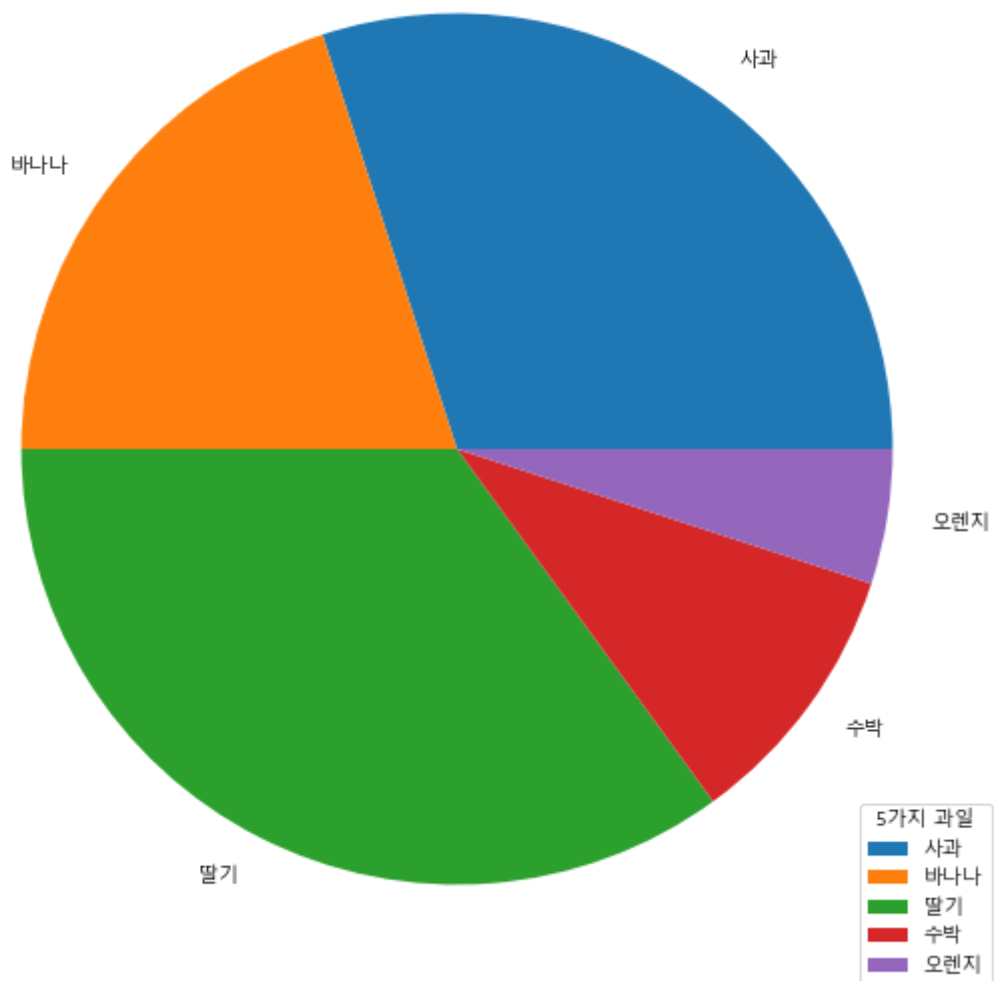
```
plt.figure(figsize=(10,10))

y = np.array([30,20,35,10, 5])
addlbl = ['사과', '바나나', '딸기', '수박', '오렌지']
myexplode = [0.2, 0, 0, 0, 0.3]

plt.pie(y, labels = addlbl)
plt.legend(title = "5가지 과일")
```

Out[48]:

<matplotlib.legend.Legend at 0x24f8eeab820>



Reference :

- https://www.w3schools.com/colors/colors_names.asp
(https://www.w3schools.com/colors/colors_names.asp)
- matplotlib colormap : <https://matplotlib.org/3.5.0/tutorials/colors/colormaps.html>
(<https://matplotlib.org/3.5.0/tutorials/colors/colormaps.html>)