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)

In [1]:

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

print(matplotlib.__version__)
```

```
3.3.2
```

05 타이틀과 레이블

In [2]:

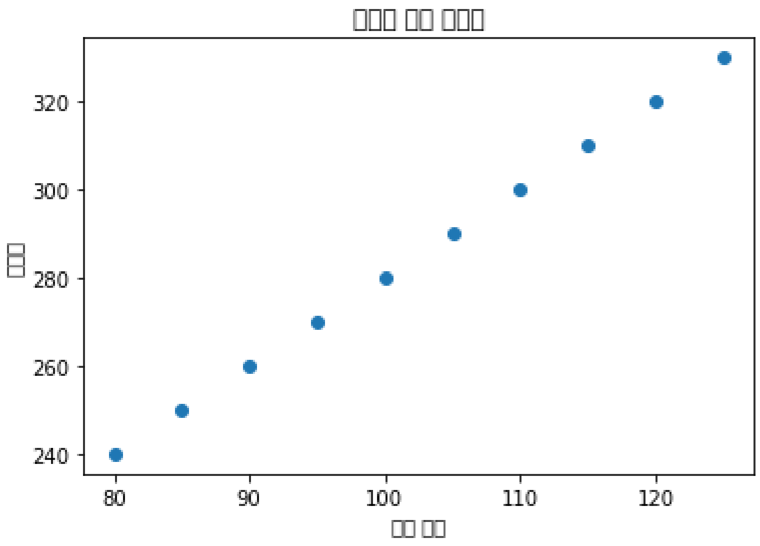
```python
import os, warnings
warnings.filterwarnings(action='ignore')
```

In [3]:

```
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[3]:

```
Text(0, 0.5, '칼로리')
```



한글 표기

In [4]:

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

In [5]:

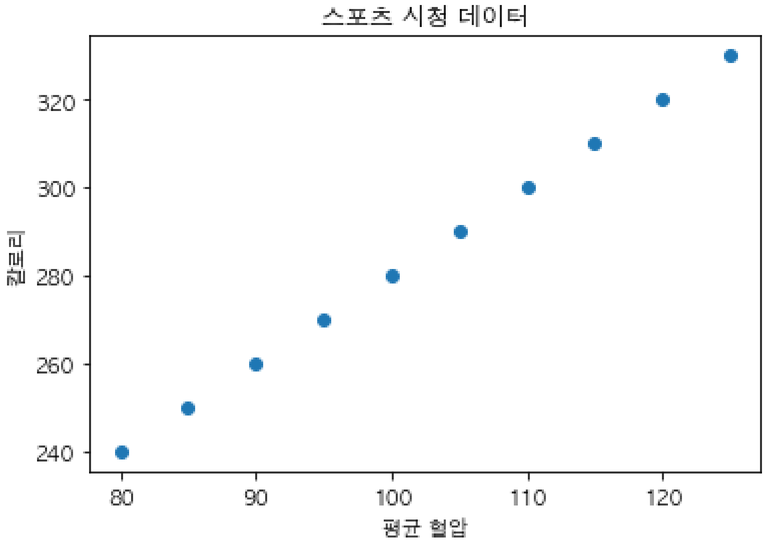
```
path = "C:/Windows/Fonts/malgun.ttf"
if platform.system() == "Windows":
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
elif platform.system()=="Darwin":
    rc('font', family='AppleGothic')
else:
    print("Unknown System")

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

In [6]:

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

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

Out[6]:

```
Text(0, 0.5, '칼로리')
```



06. Grid

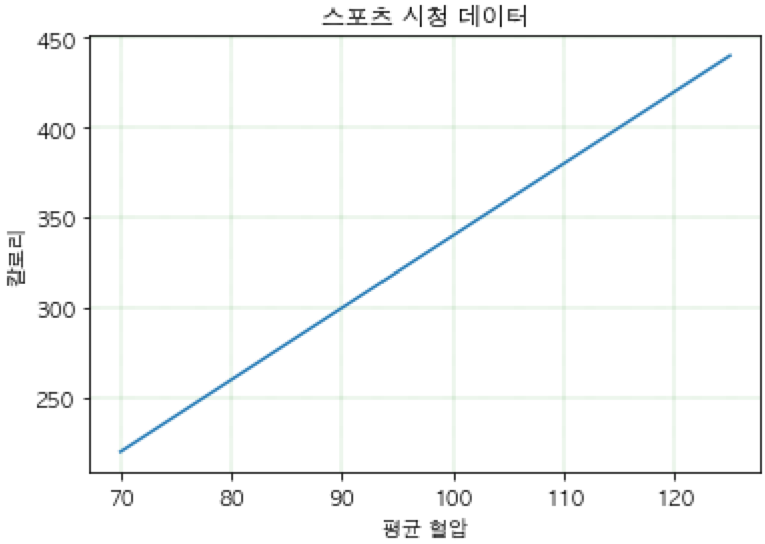
- 격자, 모눈이라는 뜻이다. 내용을 구성하는 데 사용되는 일련의 교차하는 직선 또는 곡선으로 구성된 구조.

In [7]:

```
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[7]:

```
[<matplotlib.lines.Line2D at 0x7ff8532322e0>]
```

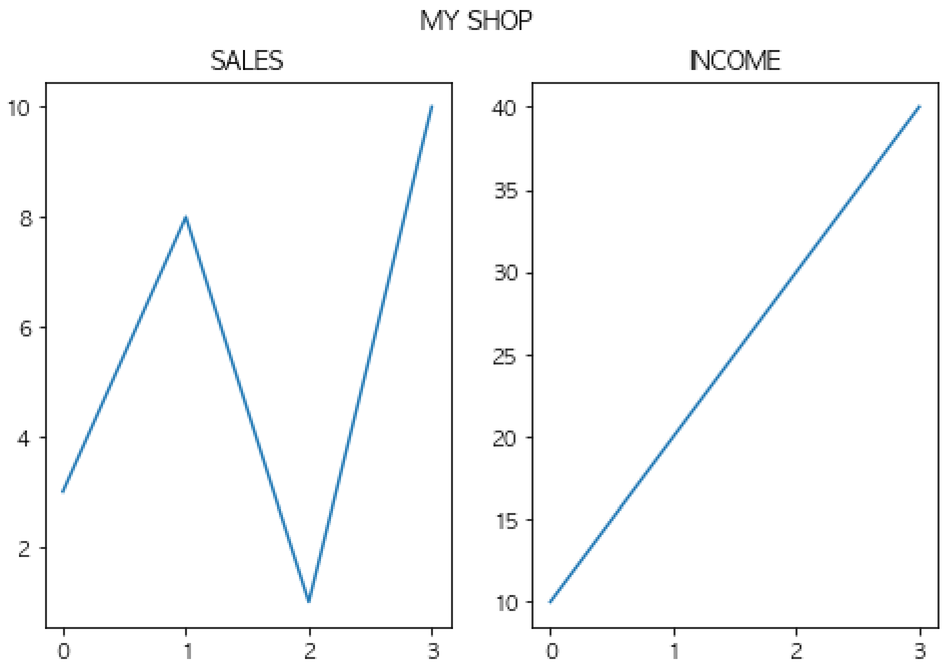


07 Super Title

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

In [8]:

```
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()
```



(실습) 2행, 2열의 그래프로 표시해 보자. 2행은 산점도 등의 다른 그래프 종류로 표시해 보

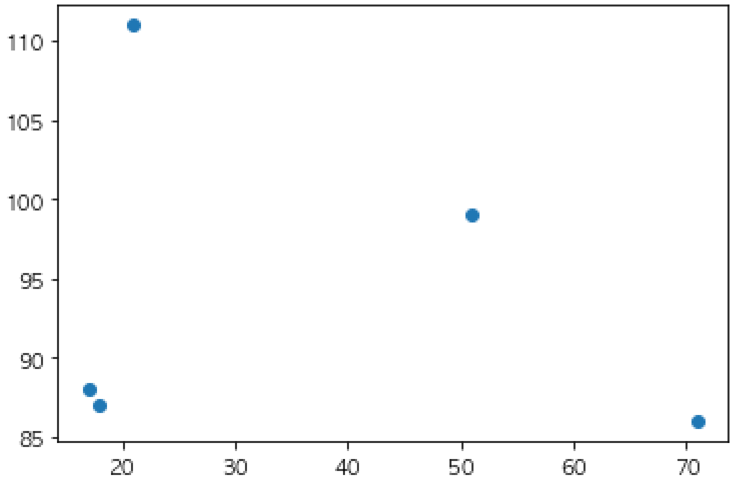
기.

08 Scatter plot

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

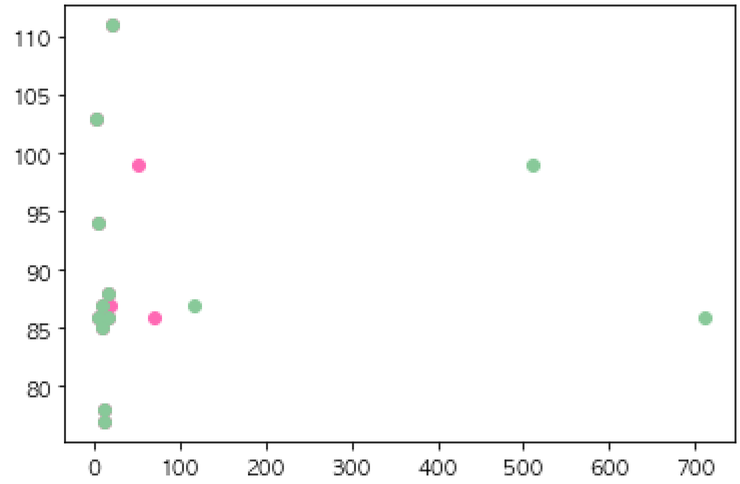
In [9]:

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

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



In [10]:

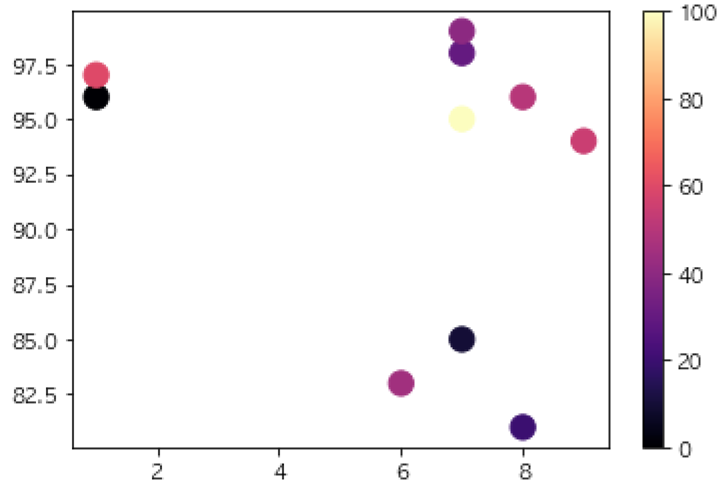
```
x = np.array([51,71,18,17,21,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

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 [11]:

```
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()
```

```
[1 7 8 7 7 6 8 9 1 7] [96 85 81 98 99 83 96 94 97 95]
10 10
```



사용가능한 colorMaps

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

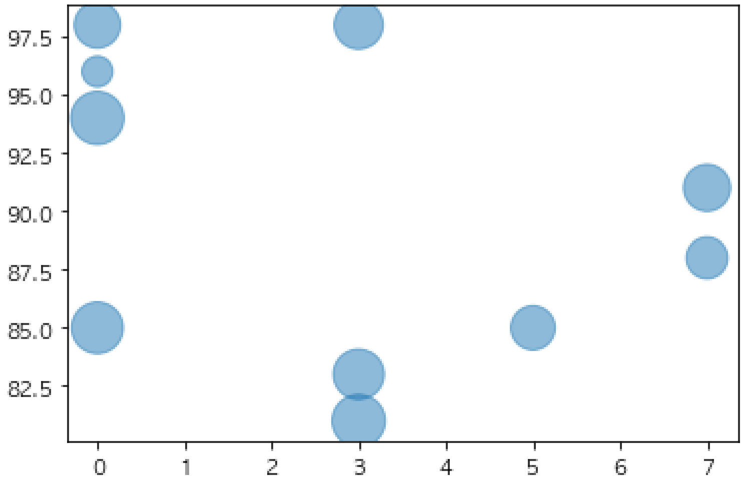
09. Alpha (투명도)

In [12]:

```
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[12]:

```
<matplotlib.collections.PathCollection at 0x7ff853ab0070>
```

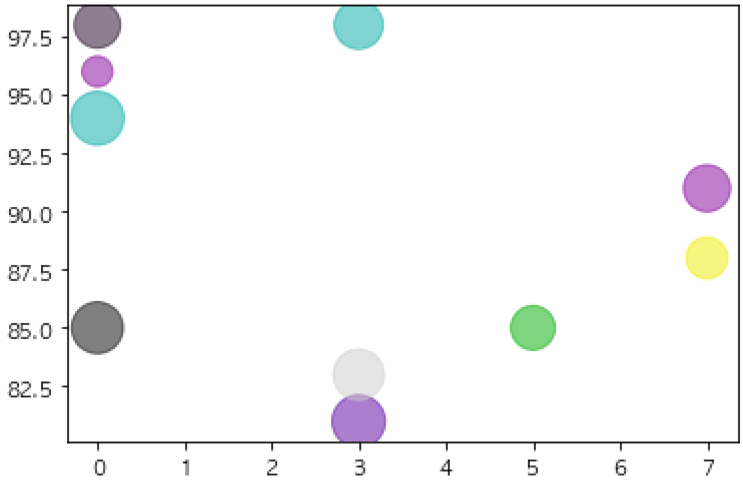


In [13]:

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

Out[13]:

```
<matplotlib.collections.PathCollection at 0x7ff853b7bd90>
```



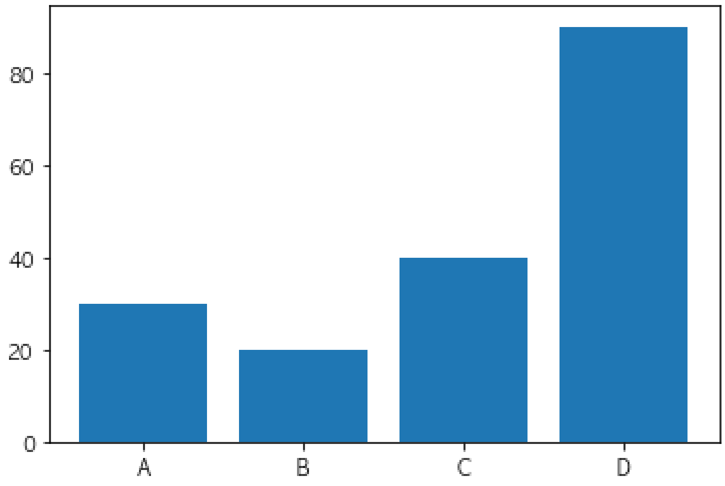
10. Bar plot

In [14]:

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

plt.bar(x, y)
```

Out[14]:

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



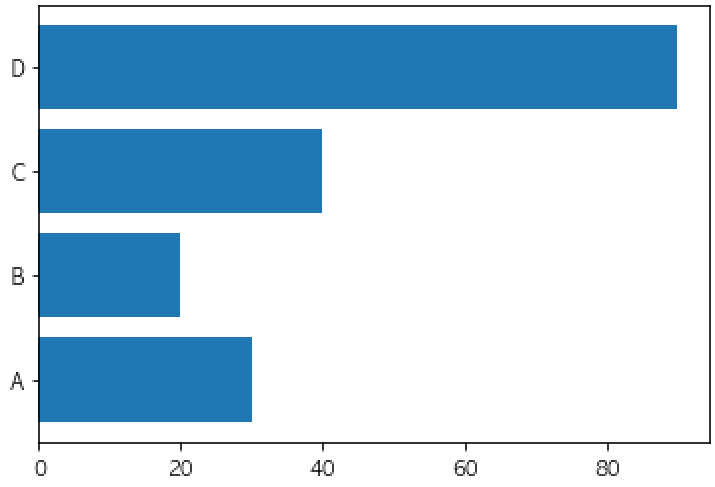
11. Horizontal Bars

In [15]:

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

plt.barh(x, y)
```

Out[15]:

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

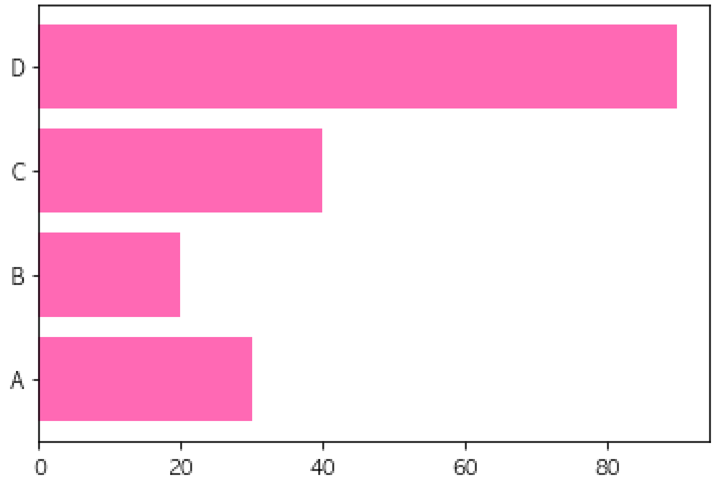


In [16]:

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

Out[16]:

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

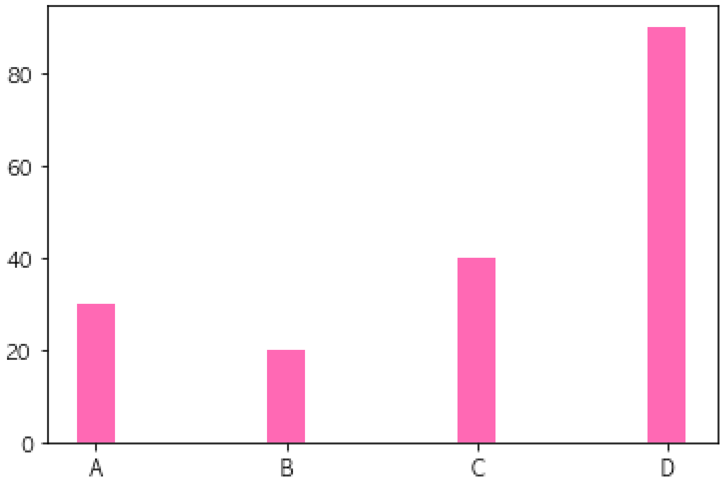


In [17]:

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

Out[17]:

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

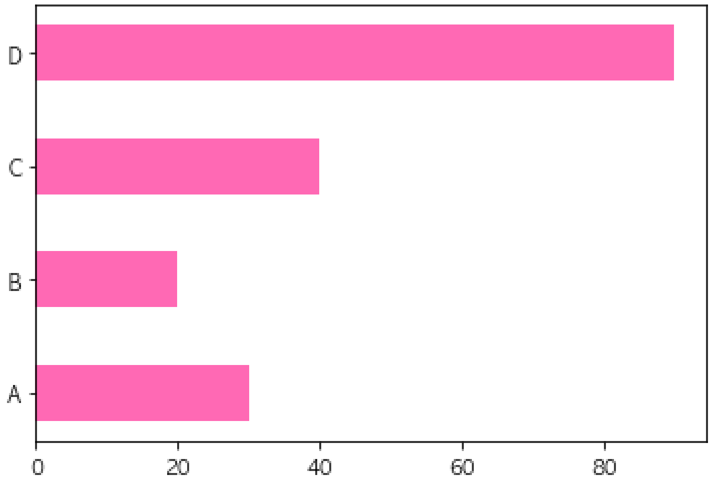


In [18]:

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

Out[18]:

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



12. 히스토그램

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

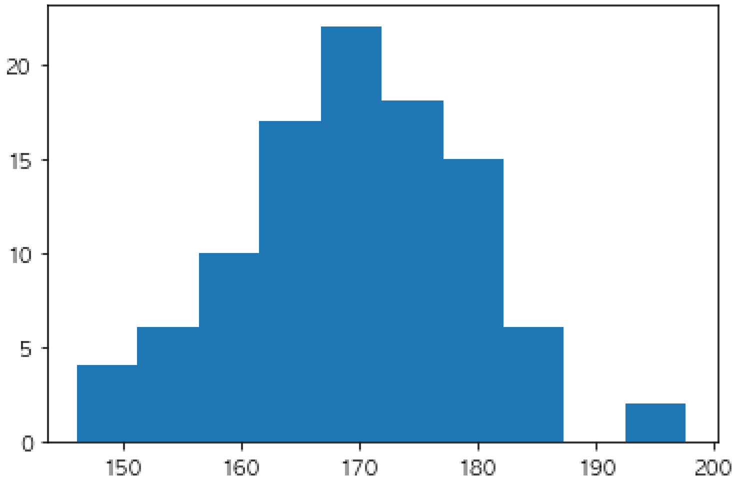
In [19]:

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

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

```
[169.18289944 169.20378913 181.87210531 162.958344   186.18770878
 162.01938502 172.00511717 165.00457592 187.33162633 173.20382454
 171.01556891 176.16974934 161.90507245 174.78202122 181.48690225
 156.73413916 180.52710102 170.35945237 177.16298178 186.04228516
 175.48488464 162.28912731 170.81241519 185.48275047 165.83989134
 153.41858721 162.2850581  159.99854674 171.8409495  175.57130177
 163.70097711 181.60677387 182.2002332  184.62060102 175.67565404
 162.83600108 171.73384    161.1408145  172.50397628 151.99515726
 158.60401945 197.71026961 164.98561983 175.92983296 166.46216551
 168.06413114 153.6762136  151.60522119 172.87059989 178.56415916
 169.20643502 176.7476882  146.14339137 147.34788937 178.66842863
 177.66268164 170.61542056 160.14447851 178.87908707 170.7815043
 152.68410717 165.10404207 180.05921348 176.54081939 167.62304668
 158.49502394 161.84356177 170.03805633 161.6328029  149.46812352
 169.26746554 178.9920154  167.2174248  179.000543   148.90826069
 157.98621823 154.73414702 179.20926258 176.19414028 193.71293601
 172.91924484 170.0430715  171.65940844 182.0417987  162.39385234
 171.07092831 187.31353307 159.74973601 173.00468558 174.83160007
 165.58563166 171.87231101 158.93242929 160.22577493 172.99435985
 168.21117551 167.8405029  169.81041797 162.92909215 175.22638897]
```

Out[19]:

```
(array([ 4.,  6., 10., 17., 22., 18., 15.,  6.,  0.,  2.]),
 array([146.14339137, 151.30007919, 156.45676702, 161.61345484,
        166.77014267, 171.92683049, 177.08351831, 182.24020614,
        187.39689396, 192.55358178, 197.71026961]),
 <BarContainer object of 10 artists>)
```



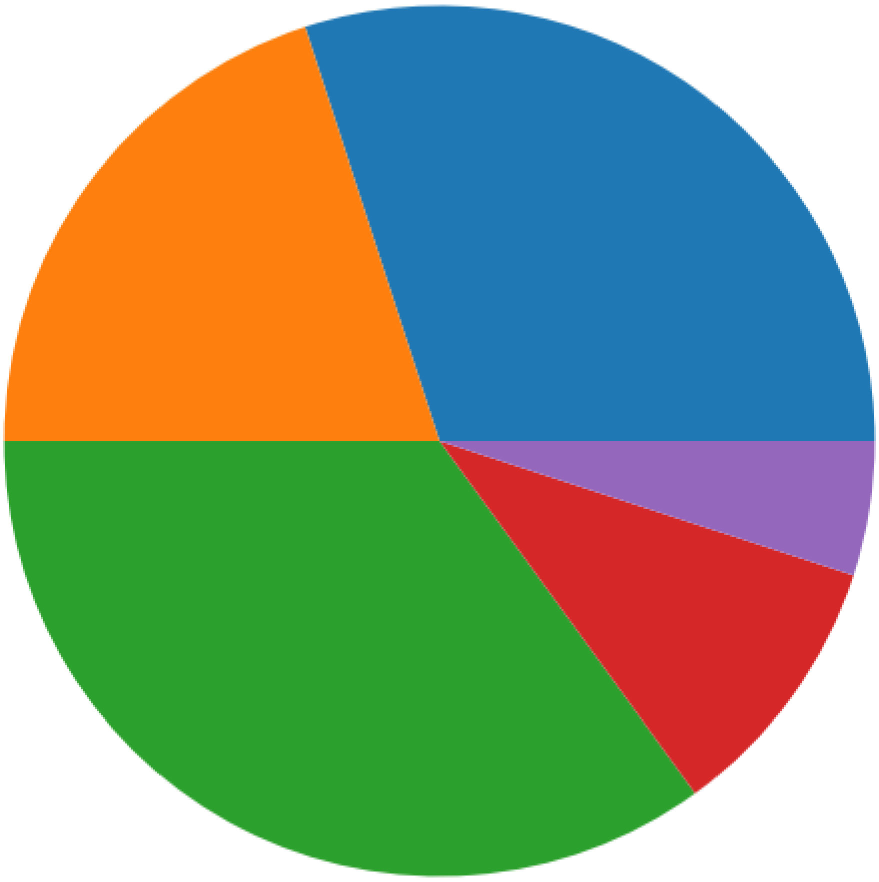
13. pie chart (원그래프)

In [20]:

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

Out[20]:

```
([<matplotlib.patches.Wedge at 0x7ff853e037f0>,
  <matplotlib.patches.Wedge at 0x7ff853e03cd0>,
  <matplotlib.patches.Wedge at 0x7ff8533e5820>,
  <matplotlib.patches.Wedge at 0x7ff85397c400>,
  <matplotlib.patches.Wedge at 0x7ff85397c880>],
 [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, '')])
```

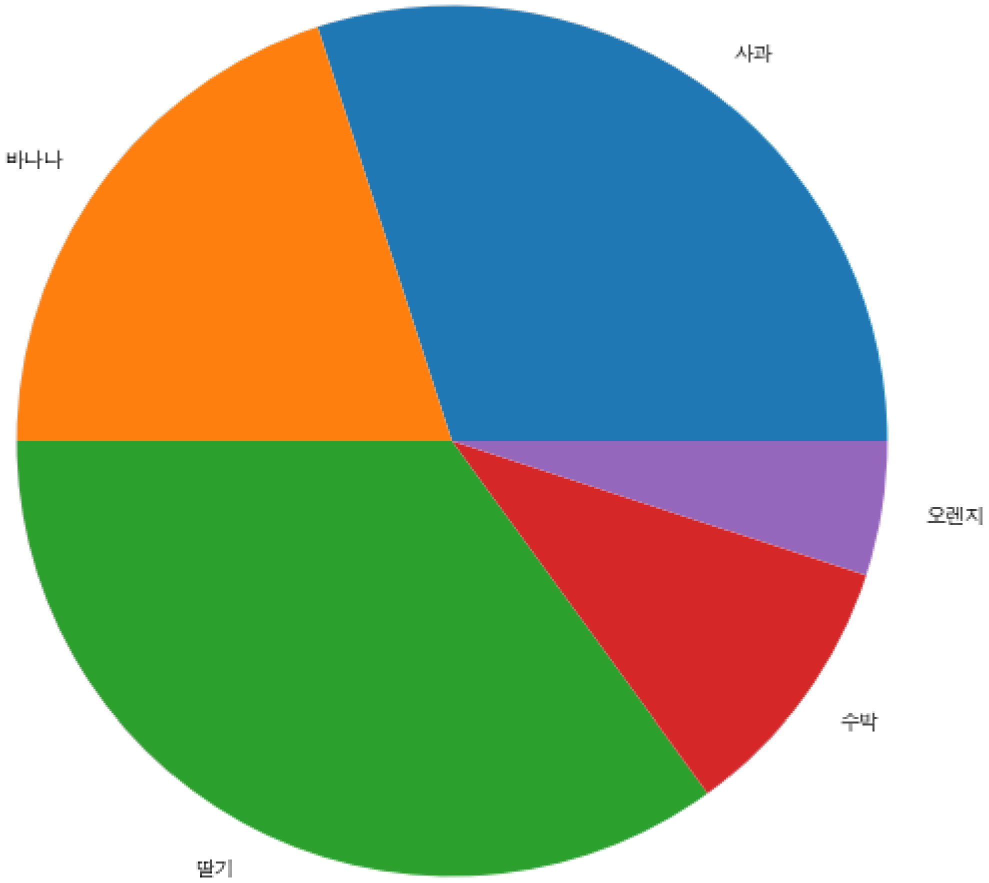


In [21]:

```
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[21]:

```
([<matplotlib.patches.Wedge at 0x7ff853e124c0>,
  <matplotlib.patches.Wedge at 0x7ff853e129a0>,
  <matplotlib.patches.Wedge at 0x7ff853e12dc0>,
  <matplotlib.patches.Wedge at 0x7ff853e1f1c0>,
  <matplotlib.patches.Wedge at 0x7ff853e1f640>],
 [Text(0.6465637441936395, 0.8899187180267094, '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')])
```

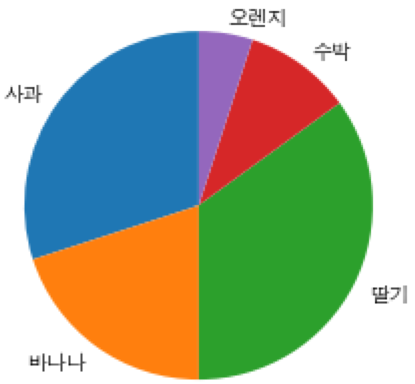


In [22]:

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

Out[22]:

```
([<matplotlib.patches.Wedge at 0x7ff8533c9f10>,
  <matplotlib.patches.Wedge at 0x7ff8533b6310>,
  <matplotlib.patches.Wedge at 0x7ff8533b6790>,
  <matplotlib.patches.Wedge at 0x7ff8533b6c10>,
  <matplotlib.patches.Wedge at 0x7ff8533e3100>],
 [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')])
```

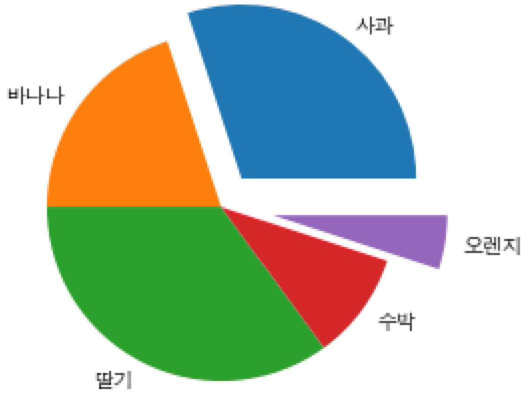


In [23]:

```
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[23]:

```
([<matplotlib.patches.Wedge at 0x7ff853e95b80>,
  <matplotlib.patches.Wedge at 0x7ff853ea30a0>,
  <matplotlib.patches.Wedge at 0x7ff853ea3520>,
  <matplotlib.patches.Wedge at 0x7ff853ea39a0>,
  <matplotlib.patches.Wedge at 0x7ff853ea3e20>],
 [Text(0.764120788592483, 1.0517221213042929, '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')])
```

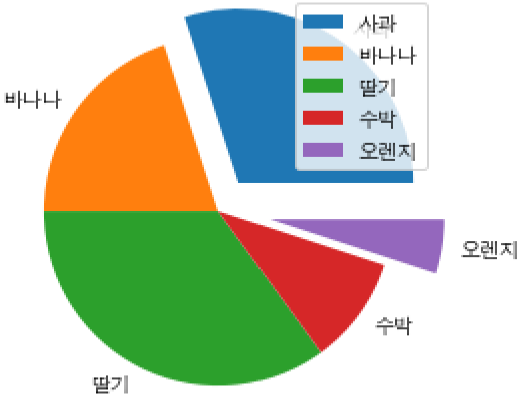


In [24]:

```
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[24]:

```
<matplotlib.legend.Legend at 0x7ff8533e3f10>
```

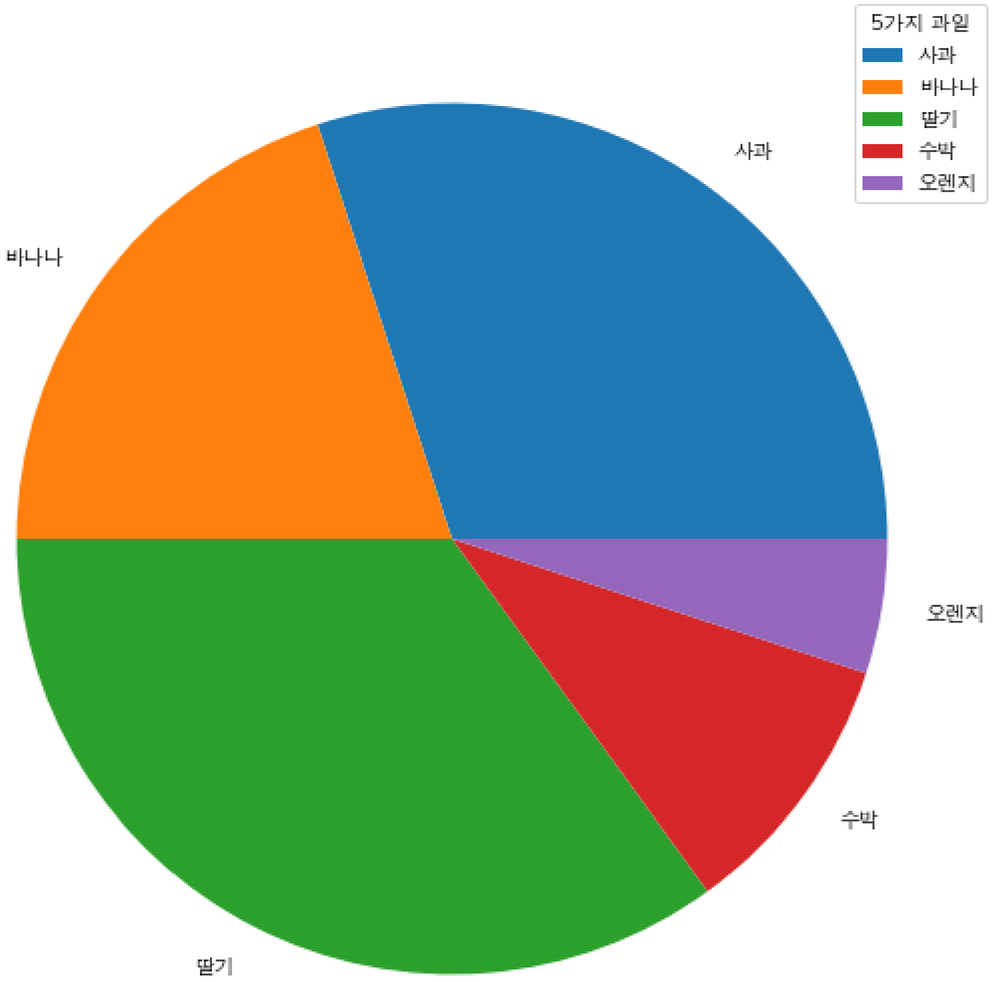


In [25]:

```
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[25]:

```
<matplotlib.legend.Legend at 0x7ff8533e3e80>
```



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)