# 파이썬을 이용한 시각화 기본

## 01 라이브러리 불러오기

In [1]:

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

- 버전 확인 : **version**

In [2]:

```
print(matplotlib.__version__)
```

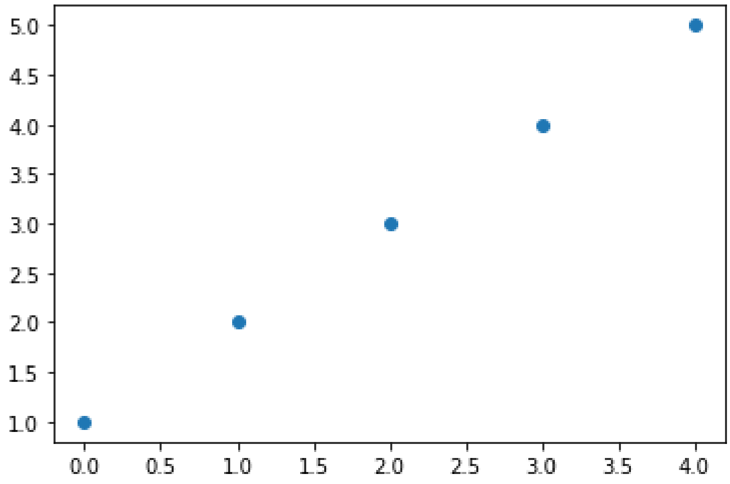
```
3.3.2
```

## 02 그래프 그려보기

- 대부분의 Matplotlib 그래프를 그리는 것이 pyplot 서브 모듈 아래에 있다. 보통 약자로 plt를 이용한다.
- plot() 함수를 사용한다.

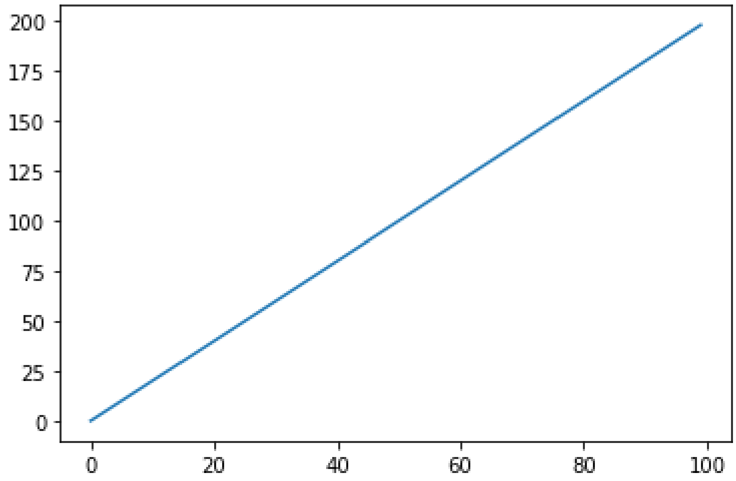
In [3]:

```
import matplotlib.pyplot as plt

plt.plot([1,2,3,4,5], 'o')  # '-' 선 : 기본값
plt.show()
```



In [4]:

```
x = range(0, 100)
y = range(0, 200,2)
plt.plot(x, y)
plt.show()
```

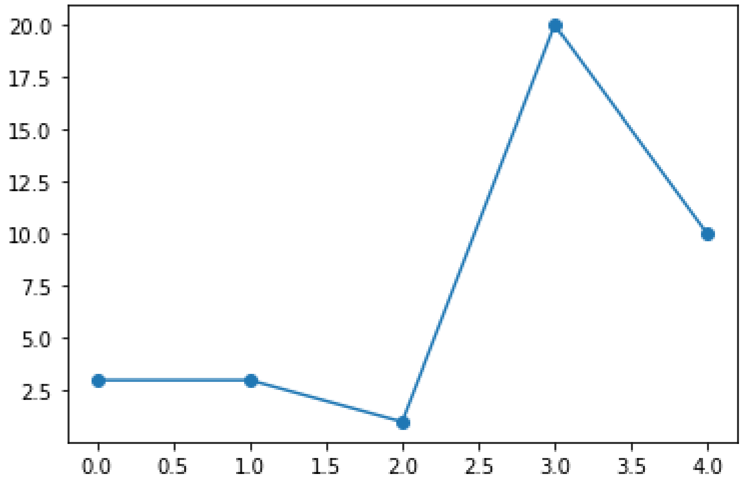


## Matplotlib Markers

- marker 인자 키워드를 사용하여 특별한 점을 강조한다.

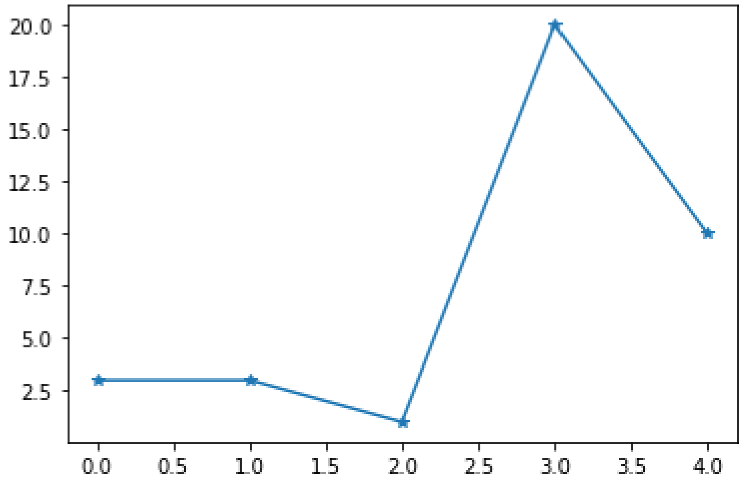
In [5]:

```
import numpy as np
ypoints = np.array([3, 3, 1, 20, 10])

plt.plot(ypoints, marker = 'o')
plt.show()
```



In [6]:

```
plt.plot(ypoints, marker = '*')
plt.show()
```

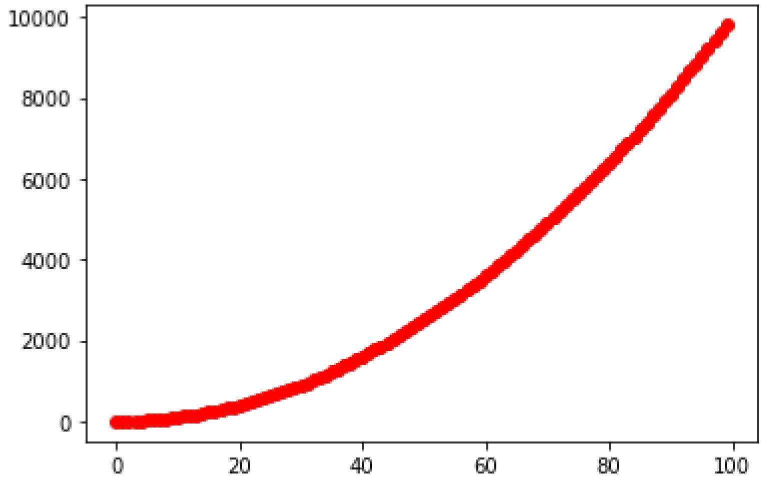


- Marker Reference
  - [https://www.w3schools.com/python/matplotlib_markers.asp (https://www.w3schools.com/python/matplotlib_markers.asp)](https://www.w3schools.com/python/matplotlib_markers.asp)

## y = x * x 의 그래프 그려보기

In [7]:

```
x = range(0, 100)
y = [ v * v for v in x ]
plt.plot(x, y, 'ro')
plt.show()
```



- 'ro'에서 'r'은 red를 의미하고, 'o'는 그래프의 마커 모양을 의미한다.

## matplot의 주요 색상

| 문자 | 색상 |
|---|---|
| b | blue(파란색) |
| g | green(녹색) |
| r | red(빨간색) |
| c | cyan(청록색) |
| m | magenta(마젠타색) |
| y | yelow(노란색) |
| k | black(검은색) |
| w | white(흰색) |

## 주요 마커

| 마커 | 의미 |
|---|---|
| o | circle(원) |
| v | triangle_down(역 삼각형) |
| ^ | triangle_up(삼각형) |
| s | square(네모) |
| + | plus(플러스) |
| . | point(점) |

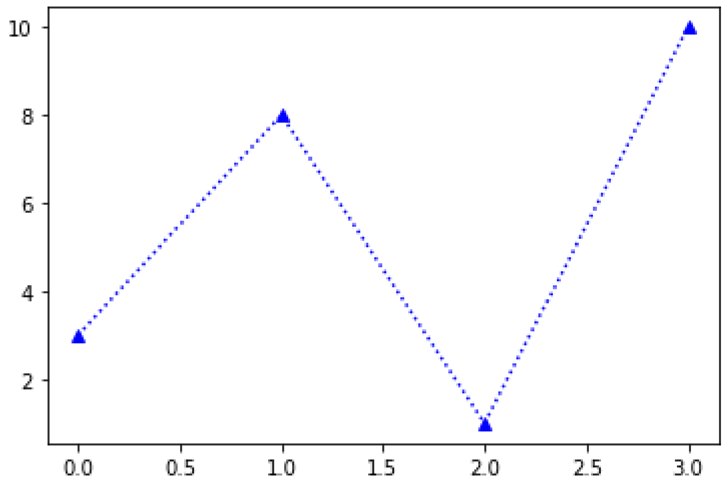
| 선 종류 | 설명 |
|---|---|
| '-' | Solid line |
| ':' | Dotted line |
| '--' | Dashed line |
| '-.' | Dashed/dotted line |

## fmt의 파라미터를 다음과 같이 사용

- marker | line | color

In [8]:

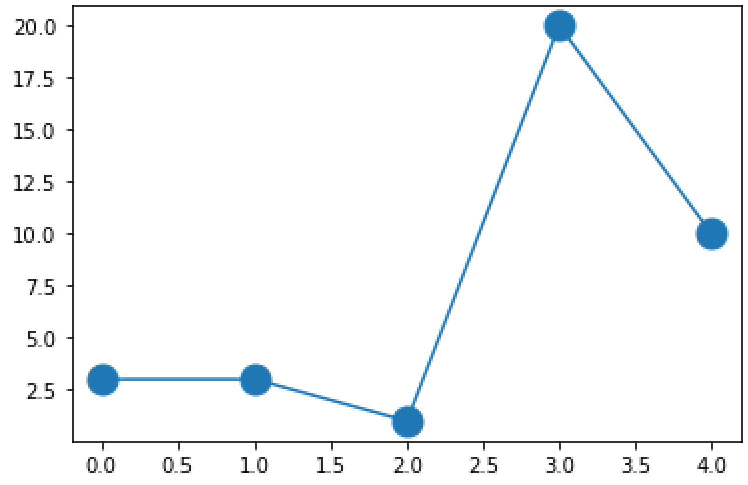
```
y = np.array([3, 8, 1, 10])

plt.plot(y, '^:b')
plt.show()
```



## 마커 사이즈

In [9]:

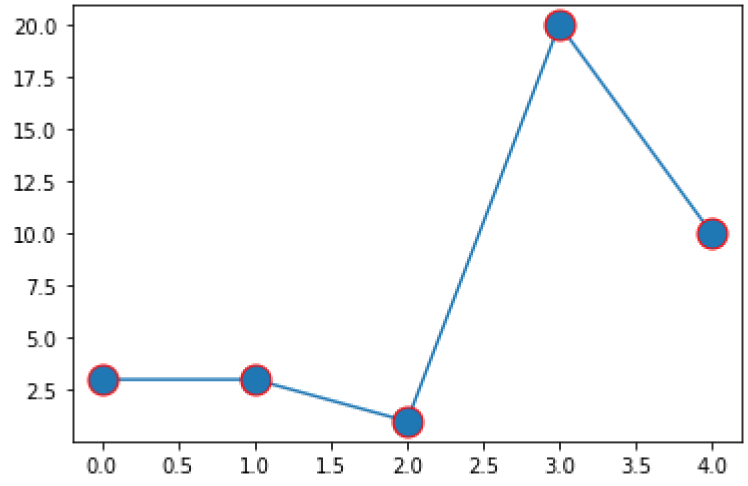
```
y = np.array([3, 3, 1, 20, 10])

plt.plot(y , marker = 'o', ms=15)
plt.show()
```



## 마커 Edge 색

In [10]:

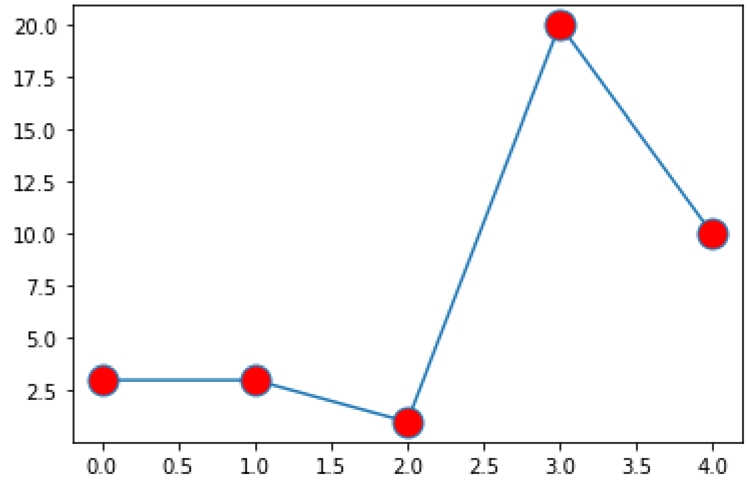
```
y = np.array([3, 3, 1, 20, 10])

plt.plot(y , marker = 'o', ms=15, mec = 'r')
plt.show()
```



## markerfacecolor

In [11]:

```
plt.plot(y , marker = 'o', ms=15, mfc = 'r')
plt.show()
```

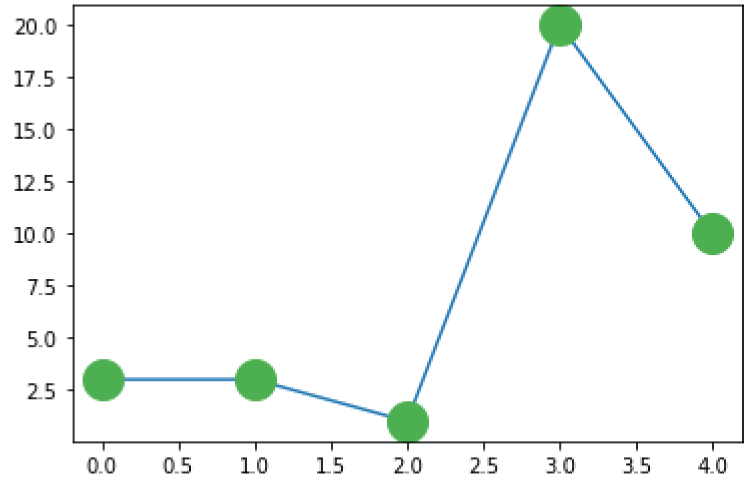


In [12]:

```
plt.plot(ypoints, marker = 'o', ms = 20, mec = '#4CAF50', mfc = '#4CAF50')
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x24f8d486d90>]
```

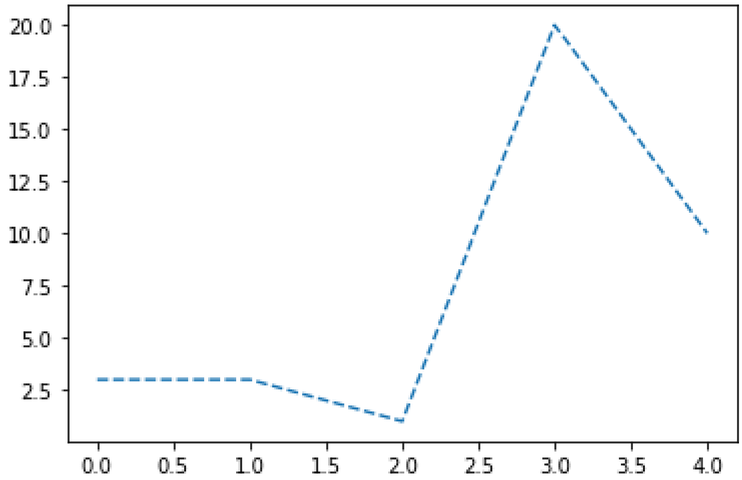


In [13]:

```
plt.plot(ypoints, linestyle = 'dashed')
```

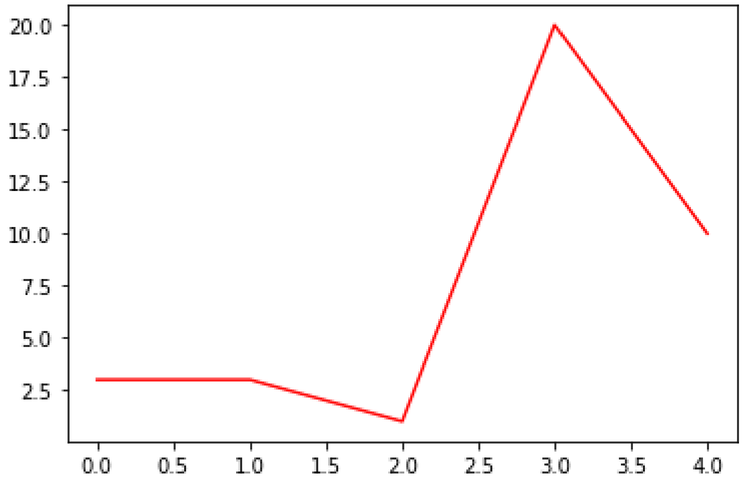
Out[13]:

```
[<matplotlib.lines.Line2D at 0x24f8d4e6d60>]
```



In [14]:

```
plt.plot(ypoints, color = 'r')
plt.show()
```

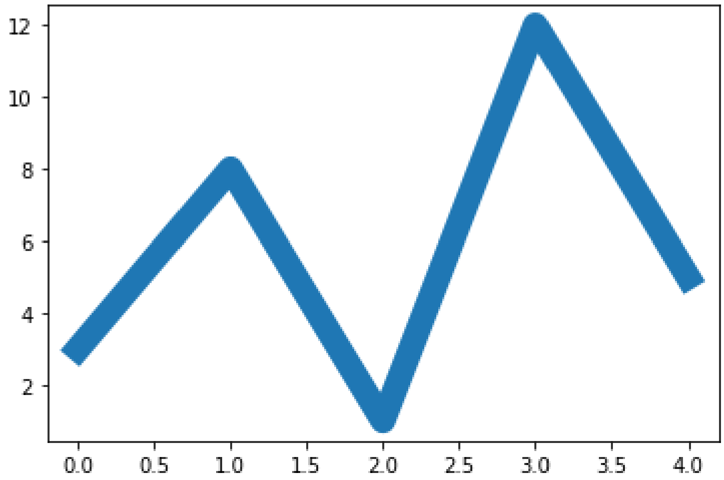


In [15]:

```
### 선 굵기
y = np.array([3, 8, 1, 12, 5])
plt.plot(y, linewidth = '12.5')
```

Out[15]:

```
[<matplotlib.lines.Line2D at 0x24f8d3f65b0>]
```

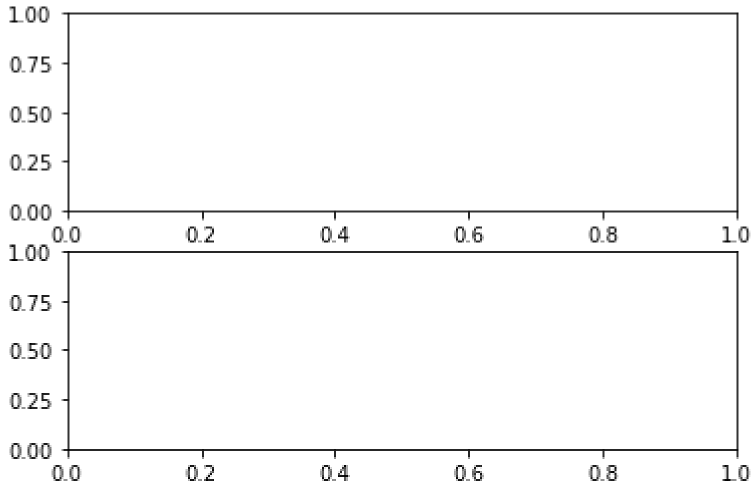


## 03 여러 개의 그래프 그리기

- 한 화면에 여러개의 그래프를 그리기 위해서는
    - (1) figure 함수를 통해 Figure 객체를 만든다.
    - (2) add_subplot 메서드를 통해 그리려는 그래프 개수만큼 subplot를 만들면 된다.

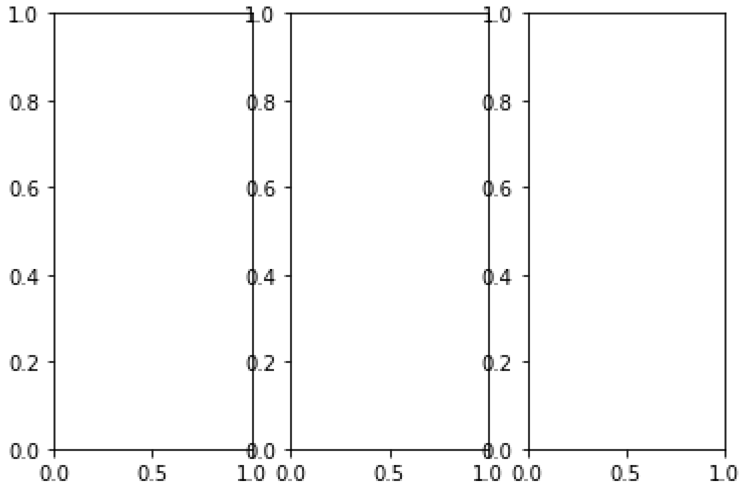
In [16]:

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



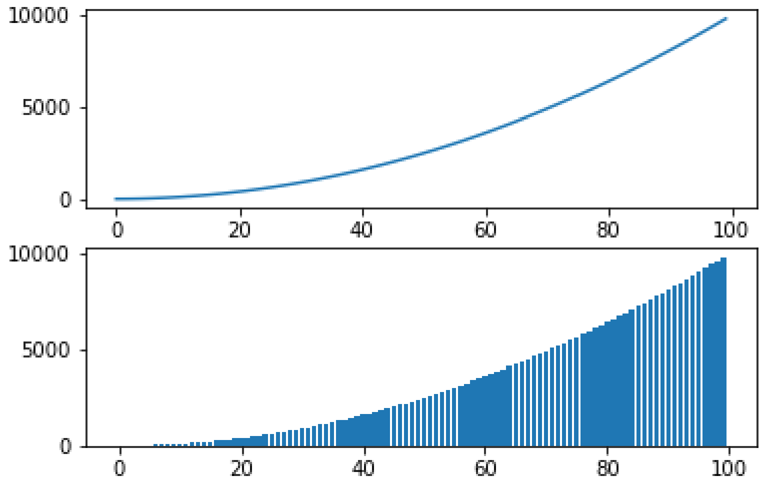
In [17]:

```
fig = plt.figure()
ax1 = fig.add_subplot(1,3,1)  # 1행 3열 의 것중에 첫번째
ax2 = fig.add_subplot(1,3,2)  # 1행 3열 의 것중에 두번째
ax3 = fig.add_subplot(1,3,3)  # 1행 3열 의 것중에 두번째
plt.show()
```



In [18]:

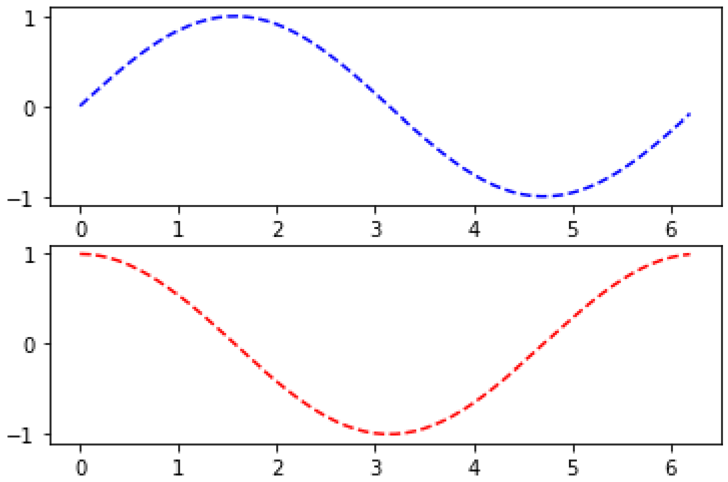
```
### plot, bar 그래프 그리기

fig = plt.figure()
ax1 = fig.add_subplot(2,1,1)
ax2 = fig.add_subplot(2,1,2)
x=range(0,100)
y=[v*v for v in x]

ax1.plot(x,y)
ax2.bar(x,y)
plt.show()
```



## sin, cos 그래프 그려보기

In [19]:

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

x = np.arange(0.0, 2 * np.pi, 0.1)
sin_y = np.sin(x)
cos_y = np.cos(x)

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

ax1.plot(x, sin_y, 'b--')
ax2.plot(x, cos_y, 'r--')

plt.show()
```



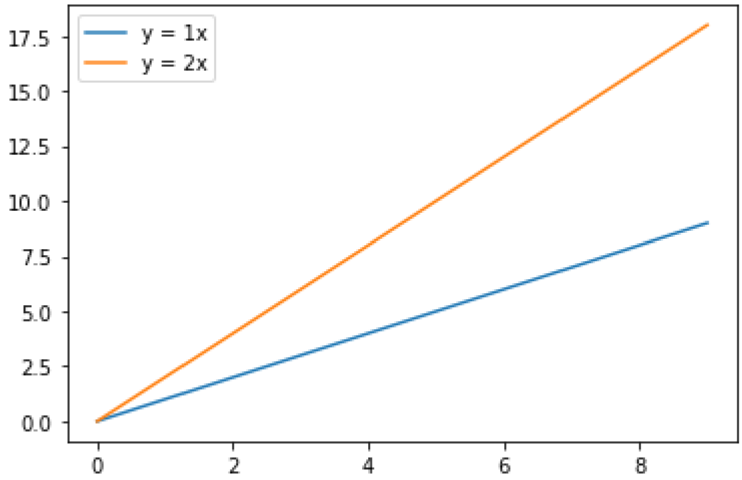
(실습2)

```
2행 2열의 그래프를 그려보자.
1행 1열: sin() 그래프
1행 2열: cos() 그래프 (표시 형식:빨간색 사각형)
2행 1열: tan() 그래프 (표시 형식:청록색 점)
2행 2열: arctan() 그래프 (표시 형식:노란색 원)
```

## 04 범례 표시 - 기본

In [20]:

```
x = np.arange(10)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.plot(x, 2*x, label='y = %ix' % 2)
plt.legend()
plt.show()
```



## 5개의 그래프 그려보기

In [21]:

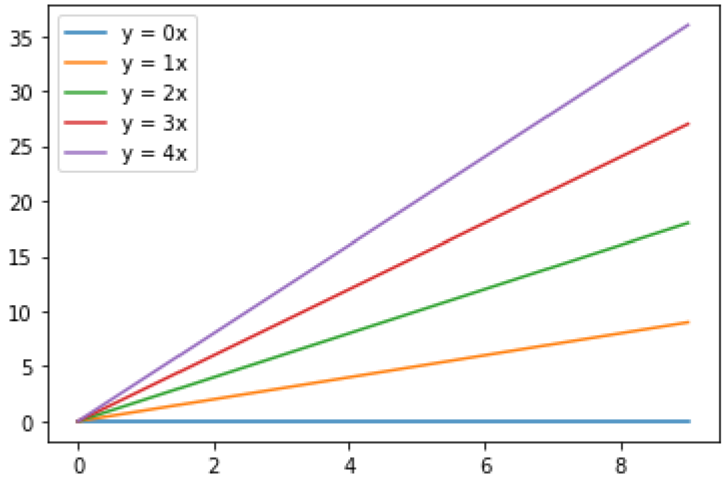
```
x = np.arange(10)
list(x)
```

Out[21]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [22]:

```
for i in range(5):
    plt.plot(x, i*x, label='y = %ix' % i)
plt.legend()
plt.show()
```



## 범례의 위치 변경과 subplots() 함수

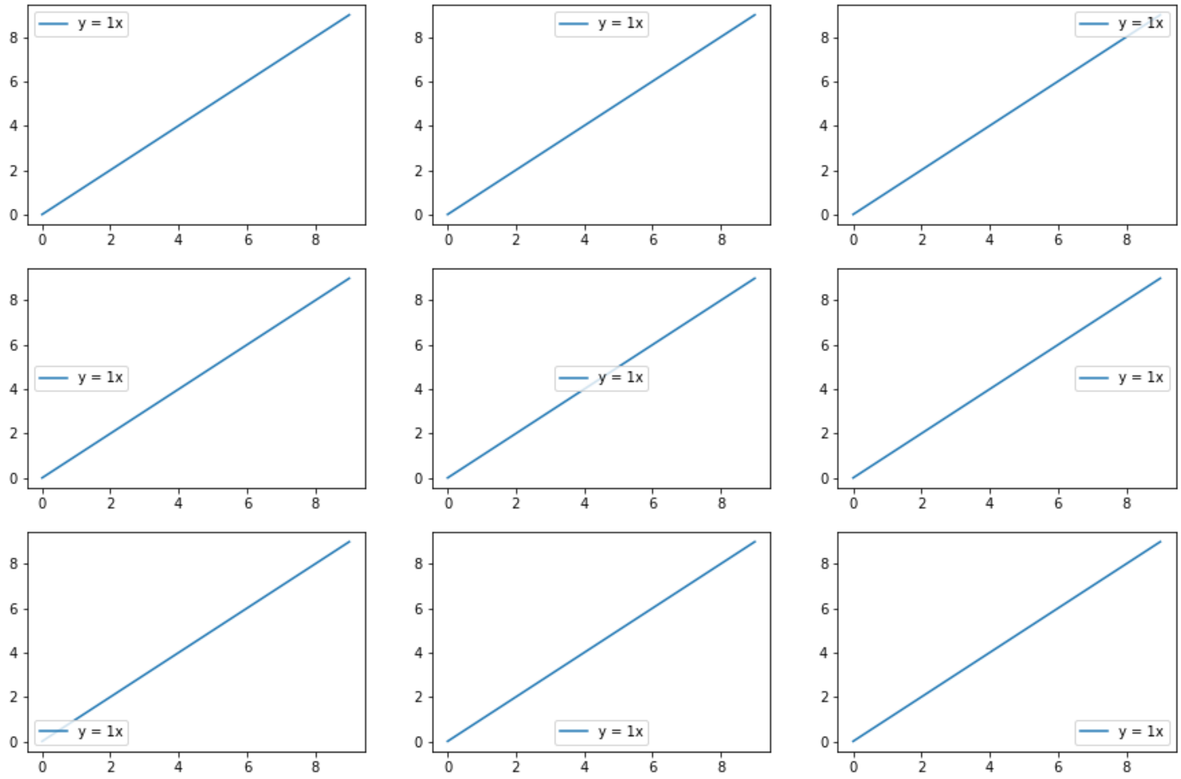
- subplots() 함수는 3개의 인수를 사용한다.

In [23]:

```
plt.figure(figsize=(15,10))

plt.subplot(3,3,1)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='upper left')    # 2

plt.subplot(3,3,2)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='upper center')  # 9

plt.subplot(3,3,3)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='upper right')  # 1

plt.subplot(3,3,4)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='center left')  # 6

plt.subplot(3,3,5)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='center')  # 10

plt.subplot(3,3,6)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='center right')  # 5

plt.subplot(3,3,7)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='lower left')  # 3

plt.subplot(3,3,8)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='lower center')  # 8

plt.subplot(3,3,9)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.legend(loc='lower right')  # 4
```

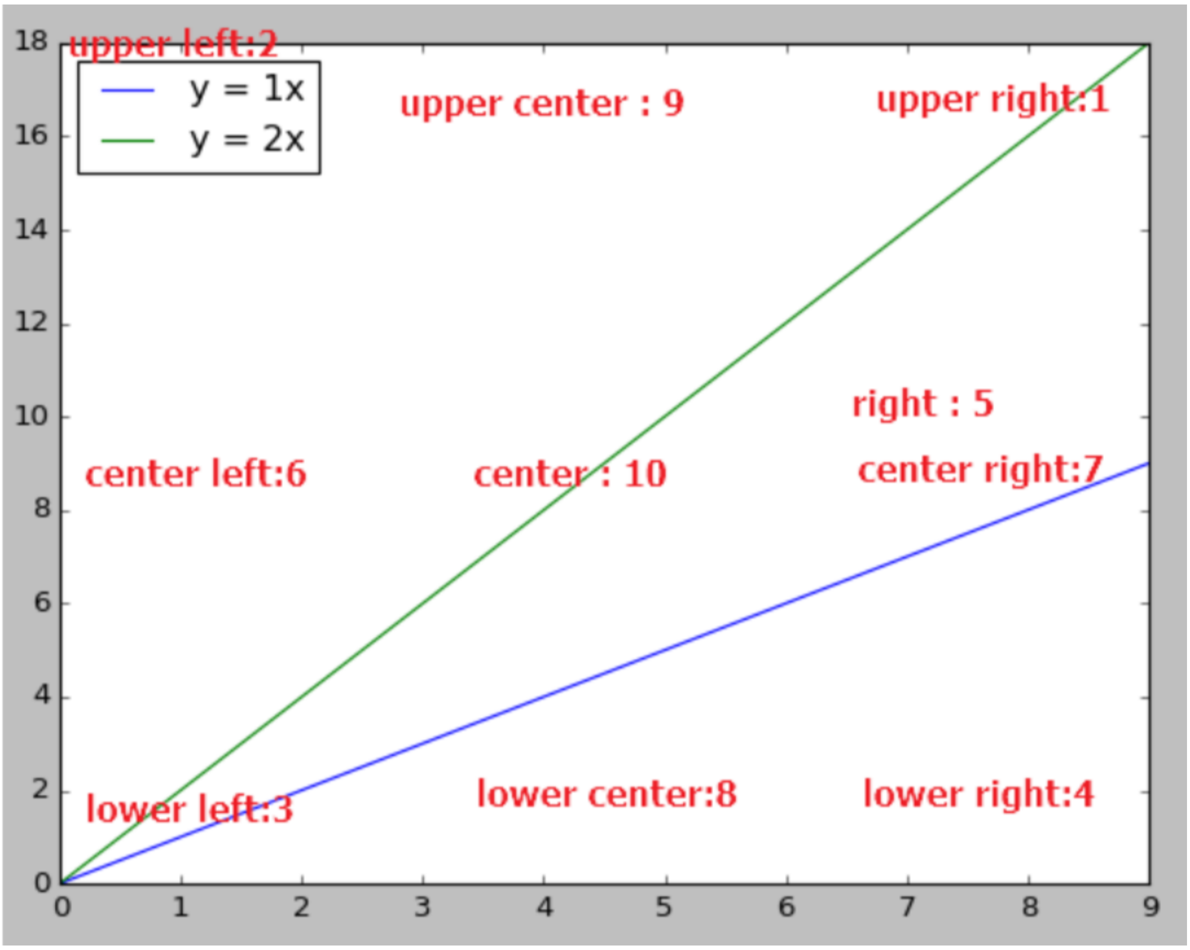
Out[23]:

```
<matplotlib.legend.Legend at 0x24f8ce67160>
```


y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8
y = 1x
0
2
4
6
8
0
2
4
6
8

In [24]:

```
import IPython.display as display
from PIL import Image

display.display(Image.open('./plt_legend_0622.png'))
```



## 05 타이틀과 레이블

In [25]:

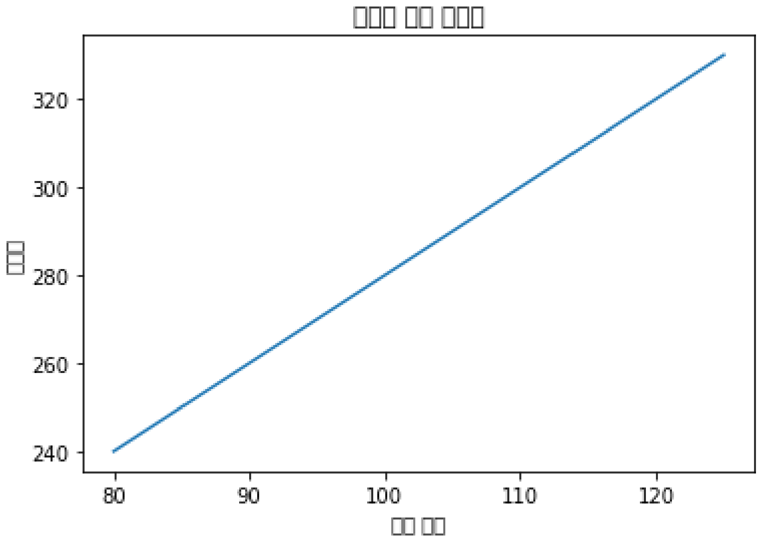
```
import os, warnings
warnings.filterwarnings(action='ignore')
```

In [26]:

```
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)

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

Out[26]:

```
Text(0, 0.5, '칼로리')
```



In [27]:

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

In [28]:

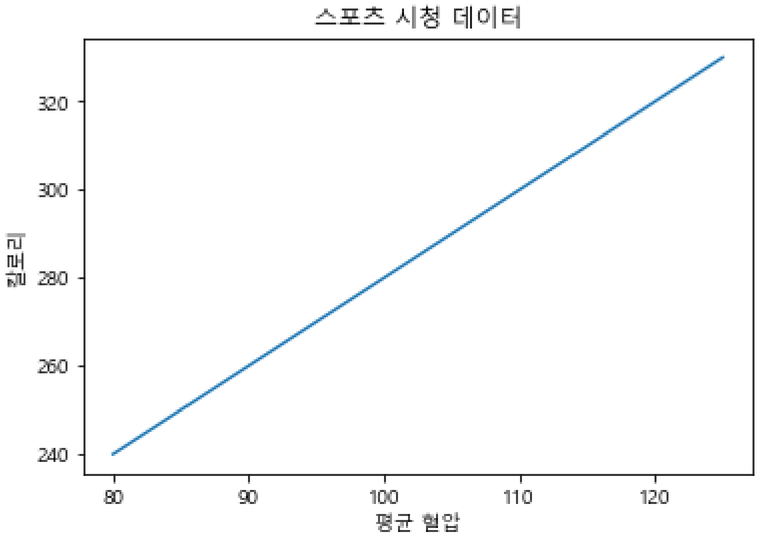
```
path = "C:/Windows/Fonts/malgun.ttf"
if platform.system() == "Windows":
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
elif platform.system()=="Darwin":
    rc('font', family='AppleGothic')
else:
    print("Unknown System")

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

In [29]:

```
plt.plot(x, y)

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

Out[29]:

```
Text(0, 0.5, '칼로리')
```



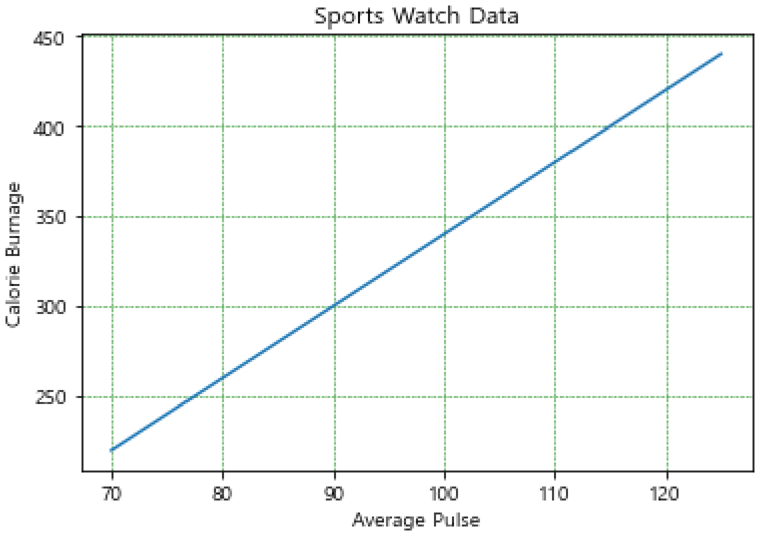
## 06 Grid

In [30]:

```
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("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)

plt.plot(x, y)
```

Out[30]:

```
[<matplotlib.lines.Line2D at 0x24f8d4ca2e0>]
```

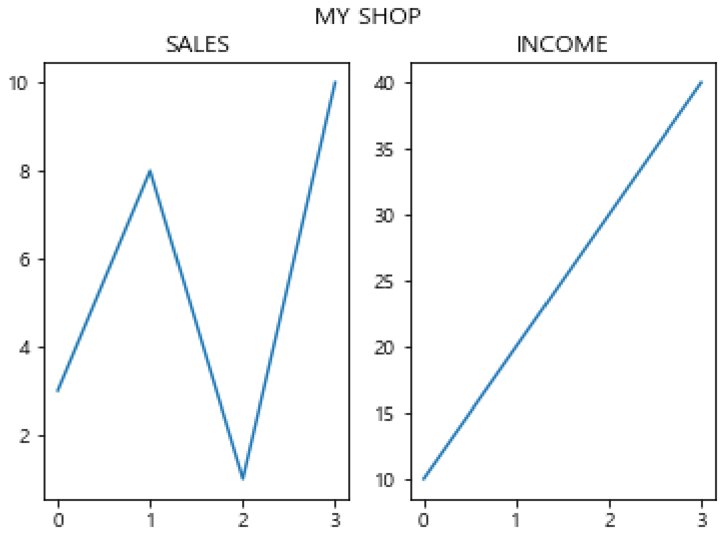


## 07 Super Title

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

In [31]:

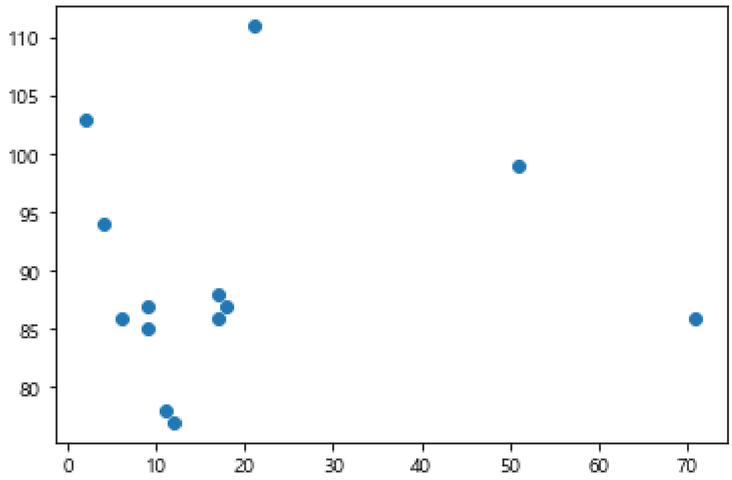
```
import matplotlib.pyplot as plt
import numpy as np

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

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])

plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("INCOME")

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



## 08 Scatter plot

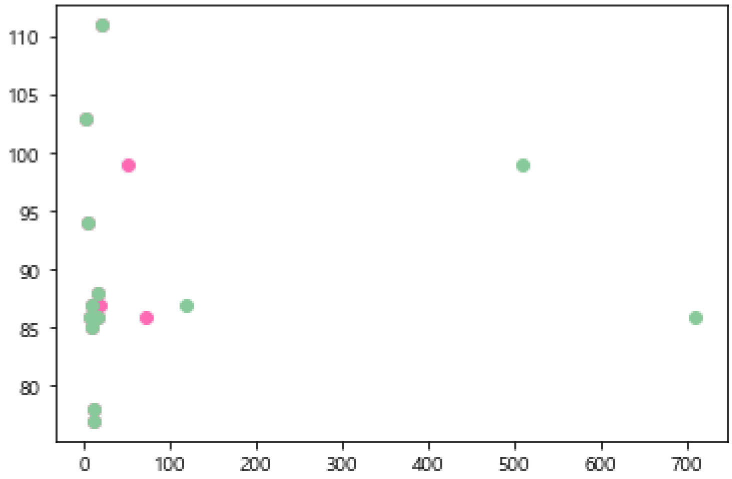
In [32]:

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

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)
plt.show()
```



In [33]:

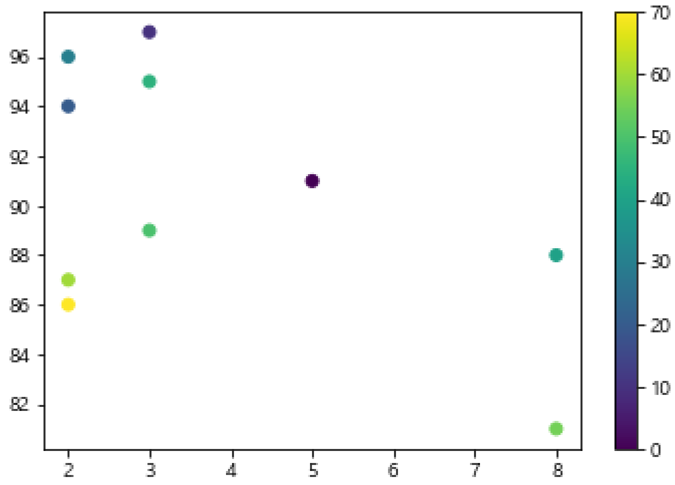
```
x = np.array([51,71,18,17,21,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

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 [34]:

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

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

colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 70])

plt.scatter(x, y, c=colors, cmap='viridis')

plt.colorbar()   # colormap 을 포함할 수 있다.

plt.show()
```

```
10 <class 'numpy.ndarray'>
10 <class 'numpy.ndarray'>
```



## 사용가능한 colorMaps

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

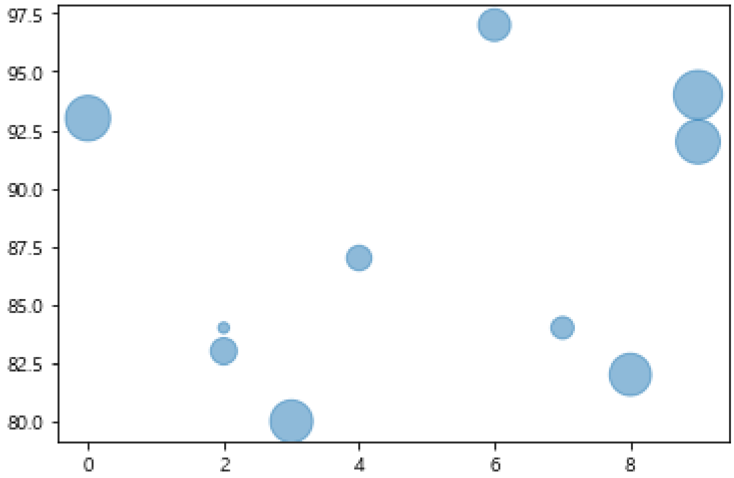
## 09. Alpha (투명도)

In [35]:

```
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[35]:

```
<matplotlib.collections.PathCollection at 0x24f8d9e62b0>
```

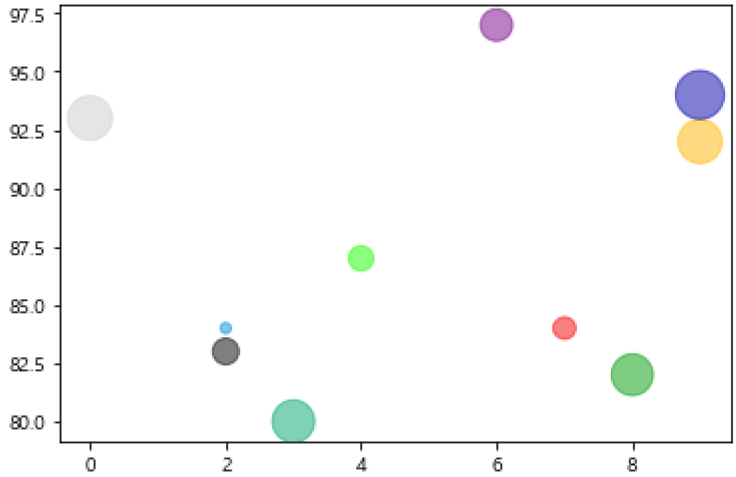


In [36]:

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

Out[36]:

```
<matplotlib.collections.PathCollection at 0x24f8da378e0>
```



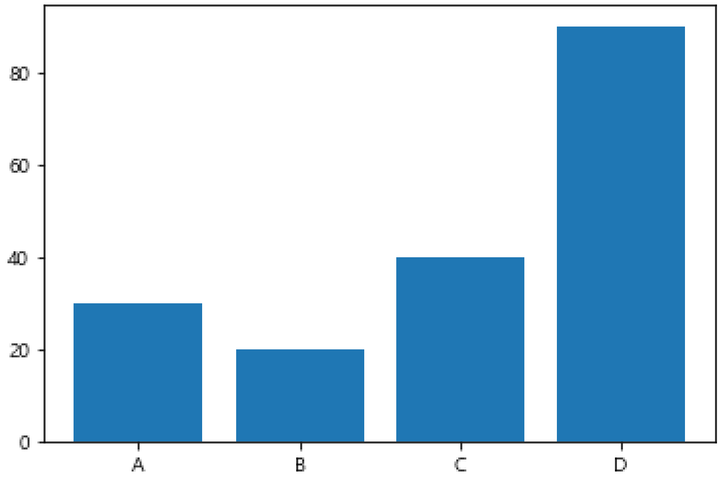
## 10. Bar plot

In [37]:

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

plt.bar(x, y)
```

Out[37]:

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



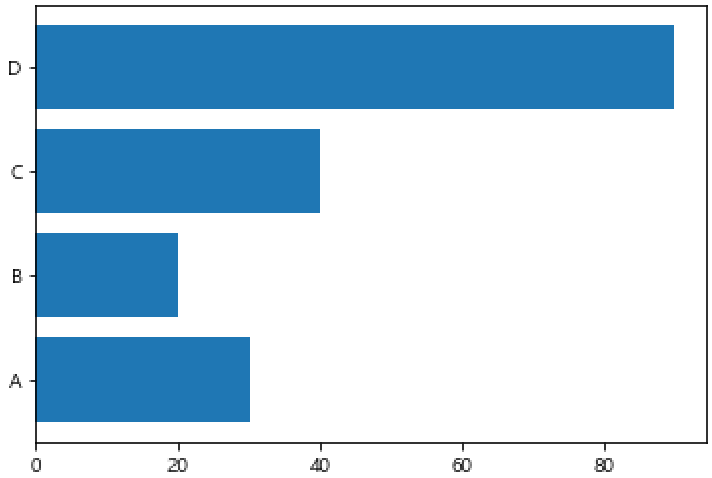
## 11. Horizontal Bars

In [38]:

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

plt.barh(x, y)
```

Out[38]:

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

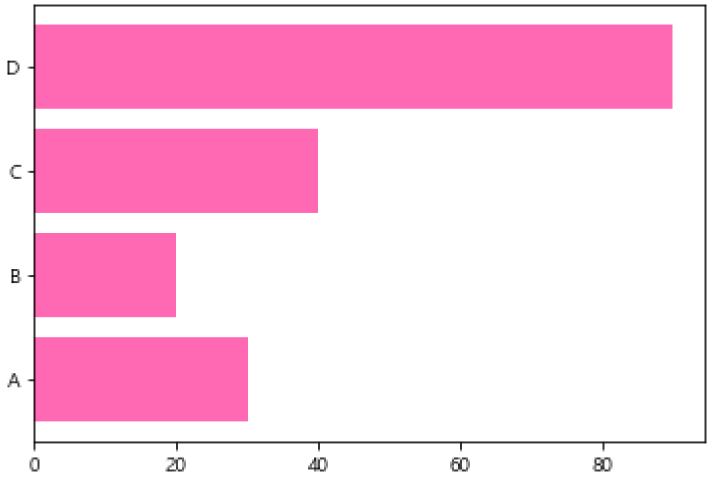


In [39]:

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

Out[39]:

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

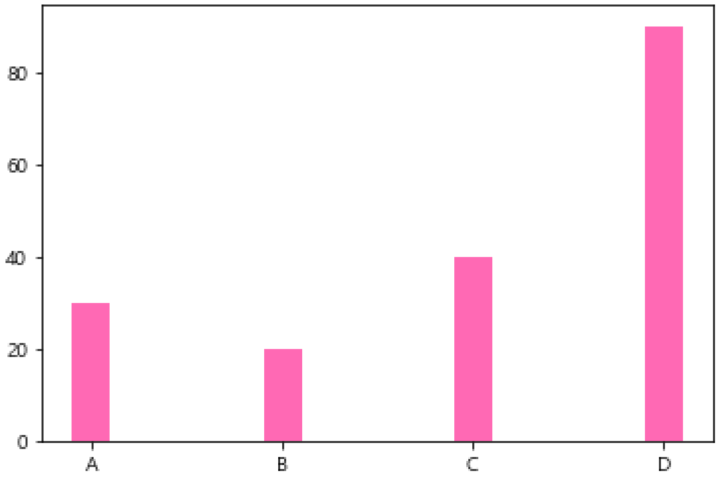


In [40]:

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

Out[40]:

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

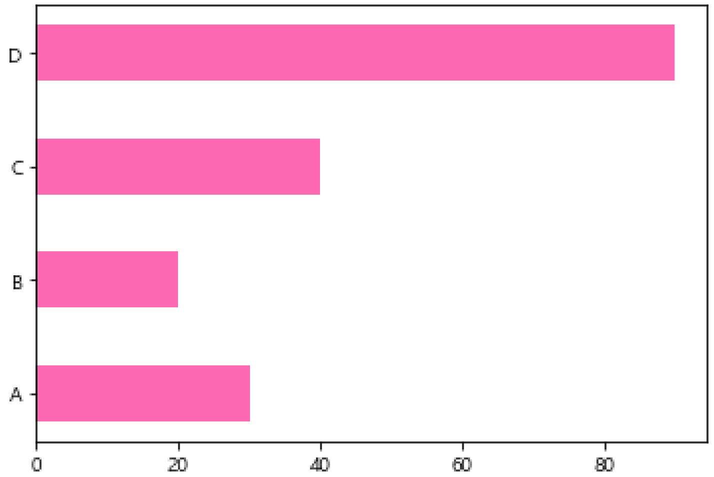


In [41]:

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

Out[41]:

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



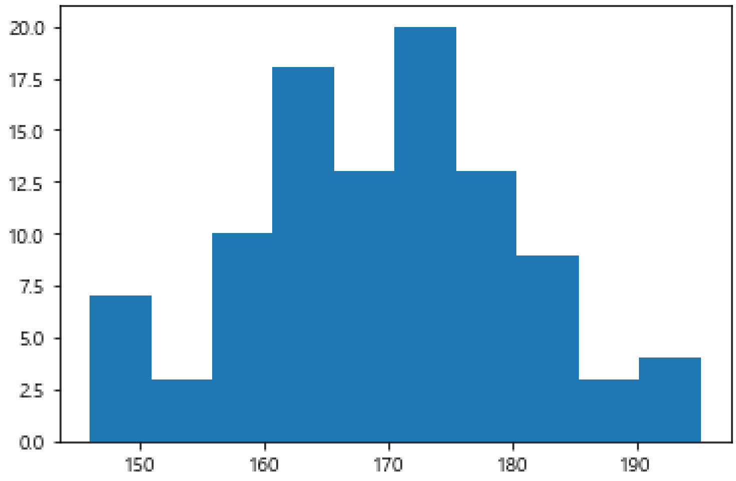
In [42]:

```
x = np.random.normal(loc=170, scale=10, size=100)  # 평균, 표준편차, 개수

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

```
[169.69458629 171.46858019 161.92977273 145.96458734 173.40034255
 194.11421523 167.53725843 176.29070351 167.10345641 147.73954935
 176.72891453 180.07661229 185.60136913 184.40691424 172.16437673
 178.67144021 180.81446499 171.92739991 173.13652684 165.58015982
 165.51446301 176.53344367 179.94643482 175.94177267 150.46839518
 171.11068525 148.00660482 176.10471659 158.3845012  168.09248767
 171.35227875 195.16474814 165.79267395 165.30464418 183.79161512
 172.39758572 163.47382438 158.23979656 184.78164331 190.9301547
 170.86683714 151.72496291 168.43298729 170.17341817 183.00137742
 164.98906376 179.03999947 149.63927331 155.94017098 189.04194649
 165.29352596 173.46352571 173.66788729 171.34059741 160.80232835
 184.8135087  164.08856646 164.83759739 147.70293621 178.60151978
 162.80414249 170.53491463 173.42079198 153.9465987  166.78345866
 171.38324495 174.08440559 159.79301597 170.73041875 167.24929541
 168.61984181 176.38011887 159.20176769 160.62007794 166.96055678
 164.47092135 155.62868559 161.0953729  176.94682741 156.43935869
 174.5076281  150.27629531 174.41699523 165.5920233  162.14316252
 163.71988669 171.71584248 165.96802666 193.13773296 160.89635957
 183.95463456 187.09905594 178.72481926 182.60641345 160.1368713
 163.57579226 158.84618868 157.86723341 174.17310489 181.03242282]
```

Out[42]:

```
(array([ 7.,  3., 10., 18., 13., 20., 13.,  9.,  3.,  4.]),
 array([145.96458734, 150.88460342, 155.8046195 , 160.72463558,
        165.64465166, 170.56466774, 175.48468382, 180.4046999 ,
        185.32471598, 190.24473206, 195.16474814]),
 <BarContainer object of 10 artists>)
```



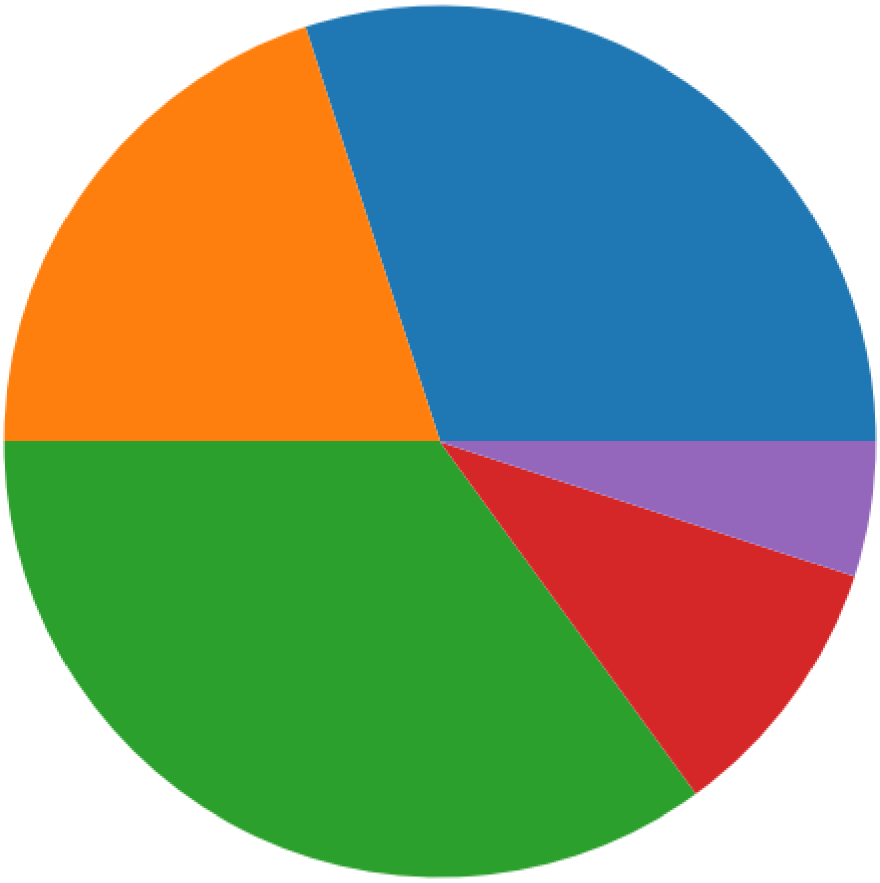
## 12. pie chart

In [43]:

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

Out[43]:

```
([<matplotlib.patches.Wedge at 0x24f8da86c40>,
  <matplotlib.patches.Wedge at 0x24f8da94100>,
  <matplotlib.patches.Wedge at 0x24f8da94580>,
  <matplotlib.patches.Wedge at 0x24f8da94a00>,
  <matplotlib.patches.Wedge at 0x24f8da94e80>],
 [Text(0.6465637441936395, 0.8899187180267095, ''),
  Text(-0.8899187482945419, 0.6465637025335369, ''),
  Text(-0.49938947630209474, -0.9801072140121813, ''),
  Text(0.8899187331606258, -0.6465637233635886, ''),
  Text(1.0864571863351944, -0.1720778377961938, '')])
```

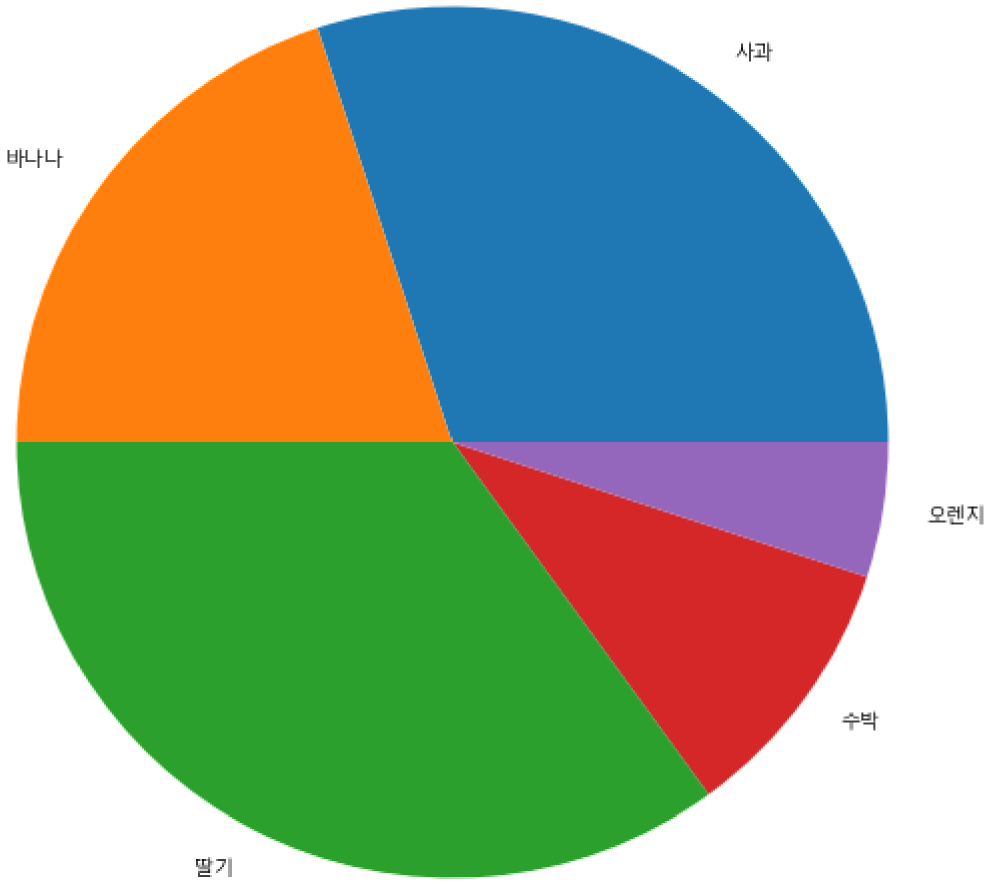


In [44]:

```
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')])
```

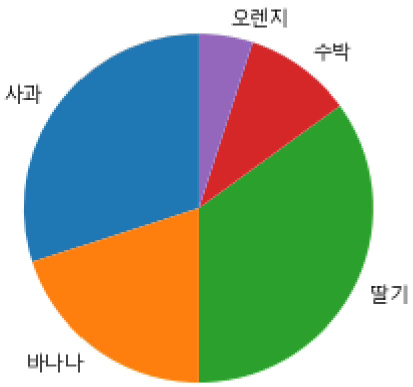


In [45]:

```
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')])
```

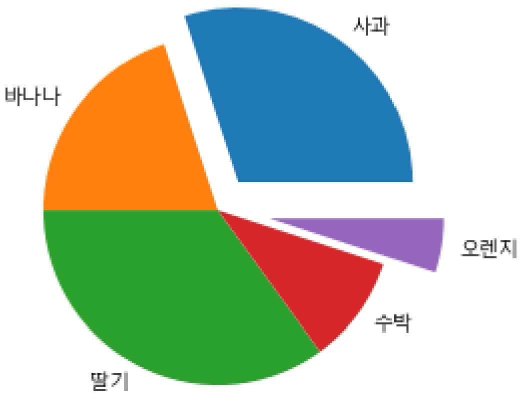


In [46]:

```
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')])
```

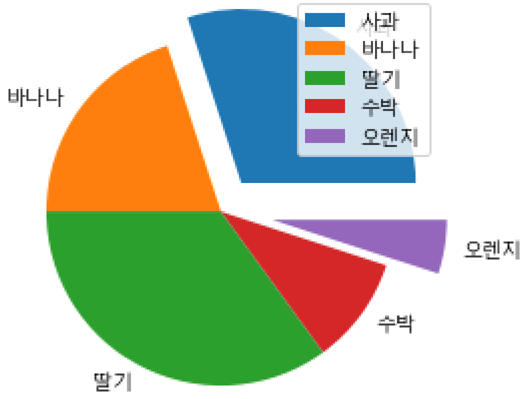


In [47]:

```
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>
```

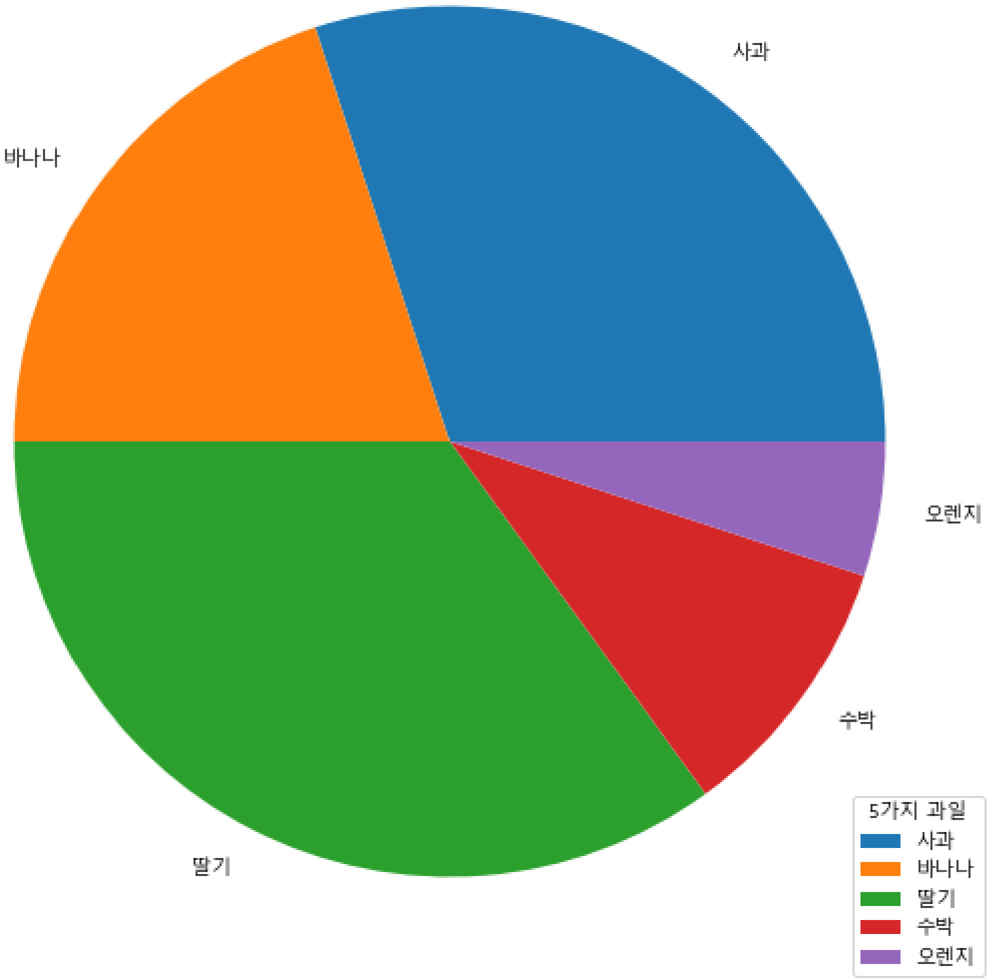


In [48]:

```
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)