

## 01 파이썬 이용한 시각화 기본

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

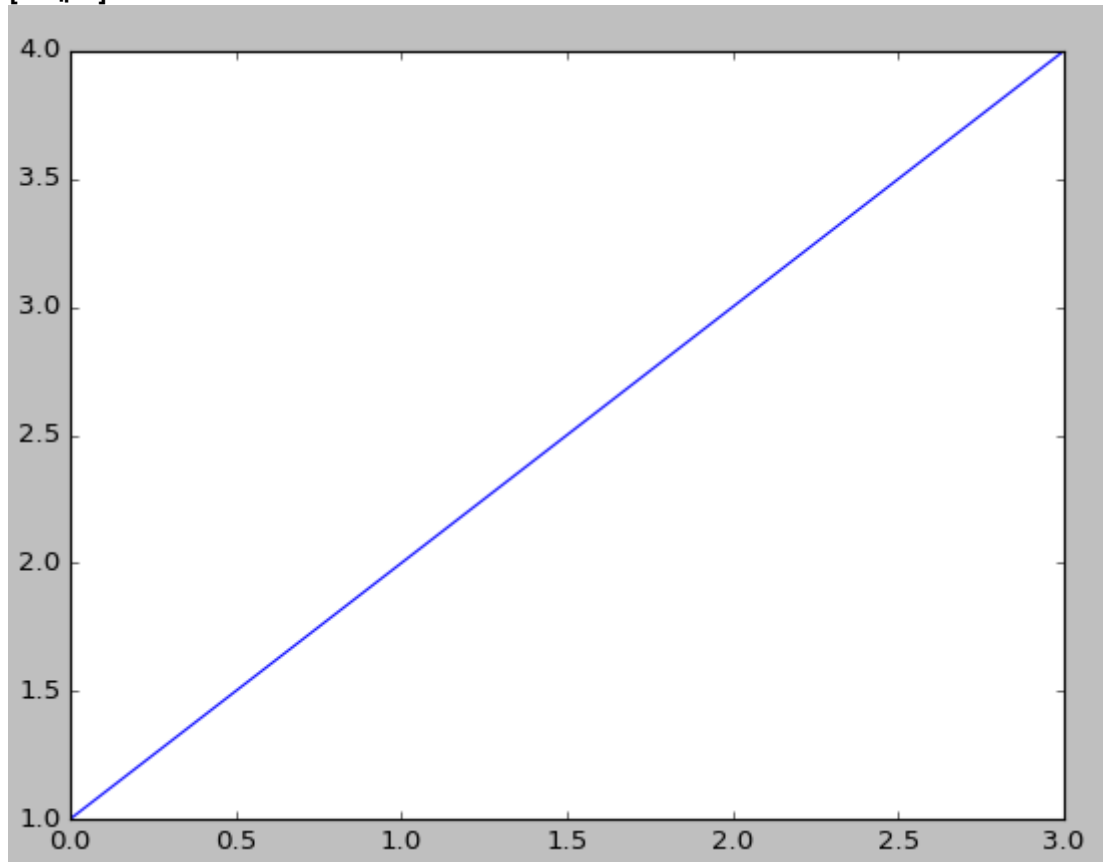
## 02 기본 그래프

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

```
def plt_plot1():  
    plt.plot([1,2,3,4])  
    plt.show()
```

plt\_plot1()

[그래프]



y축의 값 1,2,3,4를 지나는 직선을 확인

(QA)

`plt.plot([1,2,3,4])`와 같이 지정하면 어떤 값이 자동으로 입력이 될까?

나. x, y값 입력

---

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

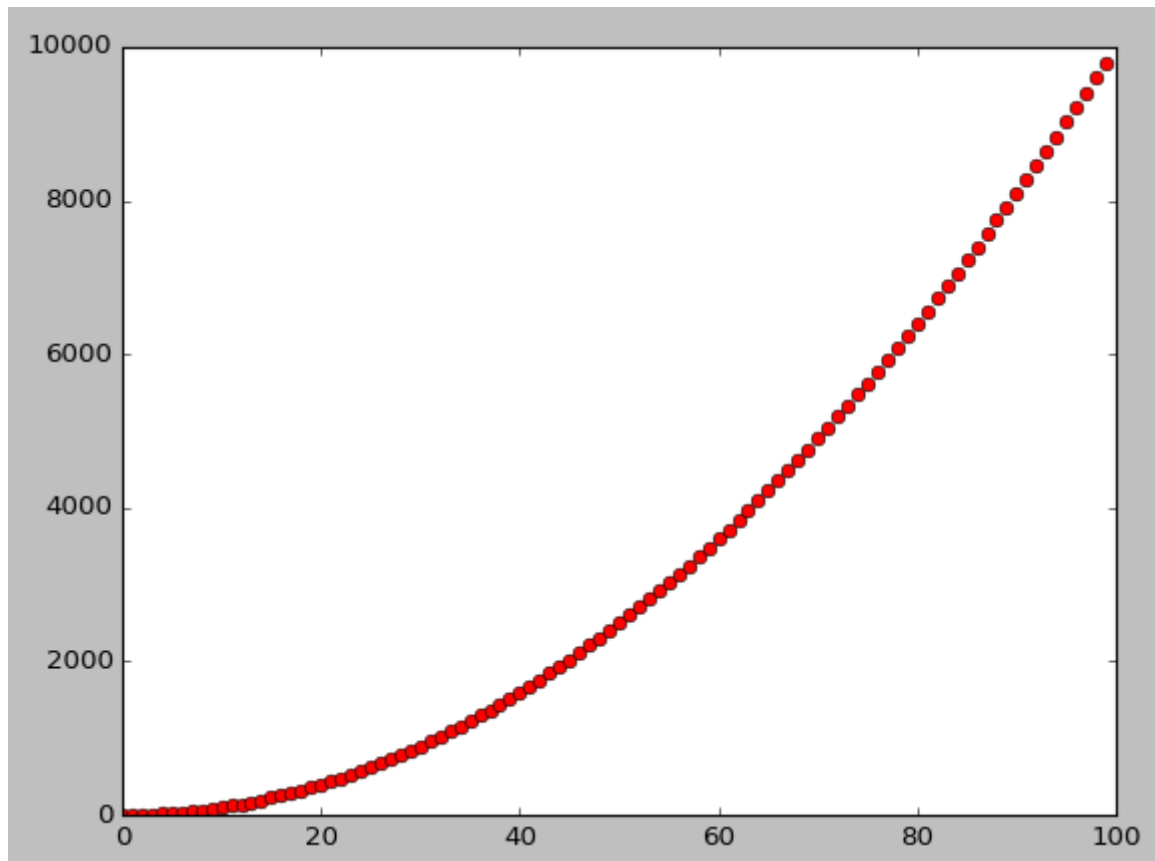
```
x = range(0, 100)
y = range(0, 200,2)
plt.plot(x, y)
plt.show()
```

---

다. x,  $y=x*x$  값 그래프 입력

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

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



**matplotlib** 주요 색상

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

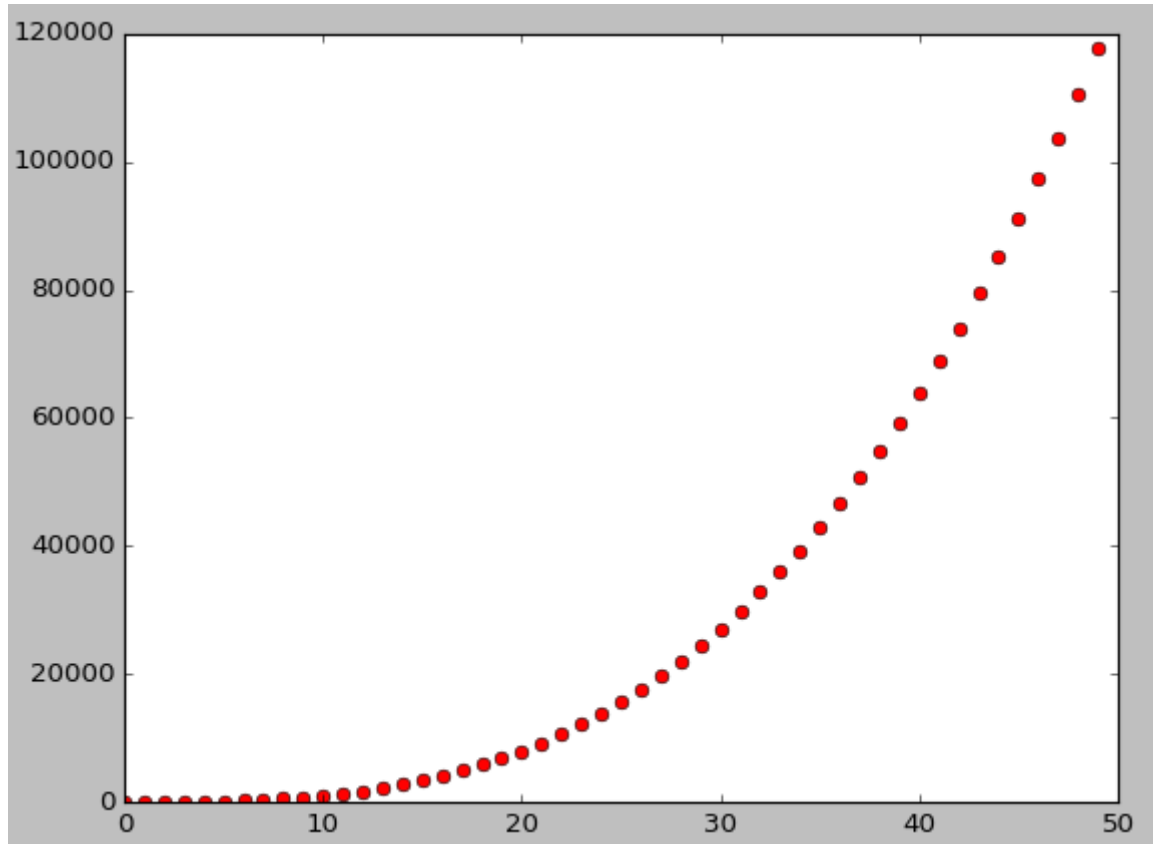
## 주요 마커

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

## (실습1)

```
x = range(0, 50)
```

$y = x \cdot x \cdot x$ 의 값을 갖는 검은색을 갖는 역삼각형 모양의 그래프를 그려보자.



### 03 여러개의 그래프 그리기

한 화면에 여러개의 그래프를 그리기 위해서는

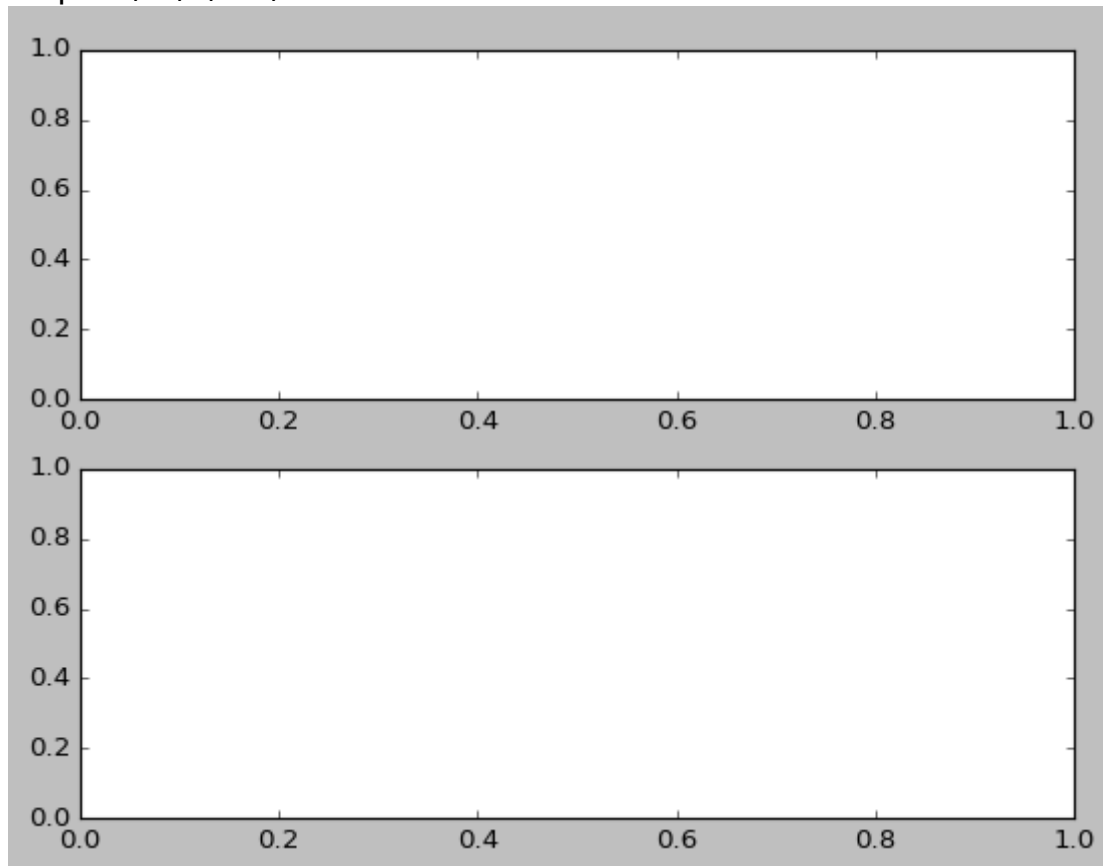
(1) figure 함수를 통해 **Figure 객체**를 먼저 만들고

(2) **add\_subplot** 메서드를 통해 그리려는 그래프 개수만큼 **subplot**를 만들면 된다.

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

Figure안에 두 개의 subplot가 존재한다.

행 방향으로 2개, 열방향으로 1개의  
subplot가 위치한다.



```
fig = plt.figure()
```

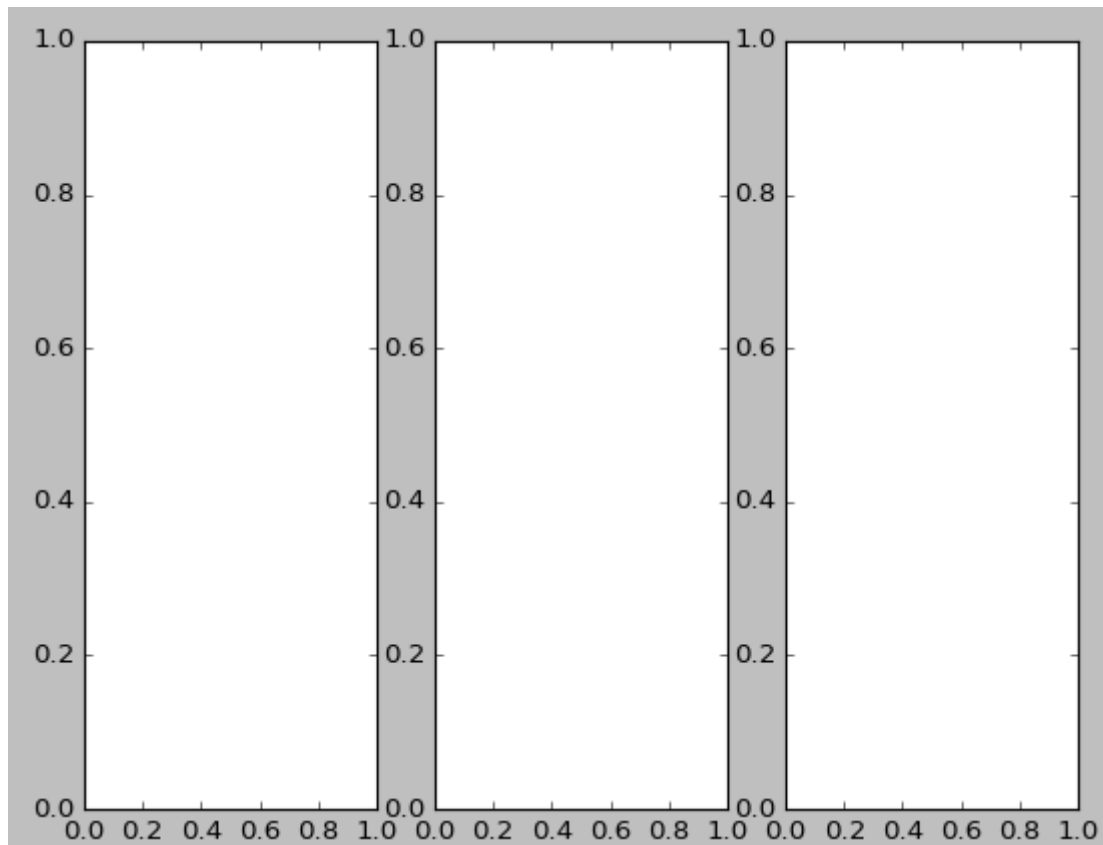
```
ax1 = fig.add_subplot(1,3,1) # 1행 3열 의 것중에 첫번째
```

```
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()
```

**fig.add\_subplot(행, 열, 순서)**



## 여러개의 그래프(plot, bar)

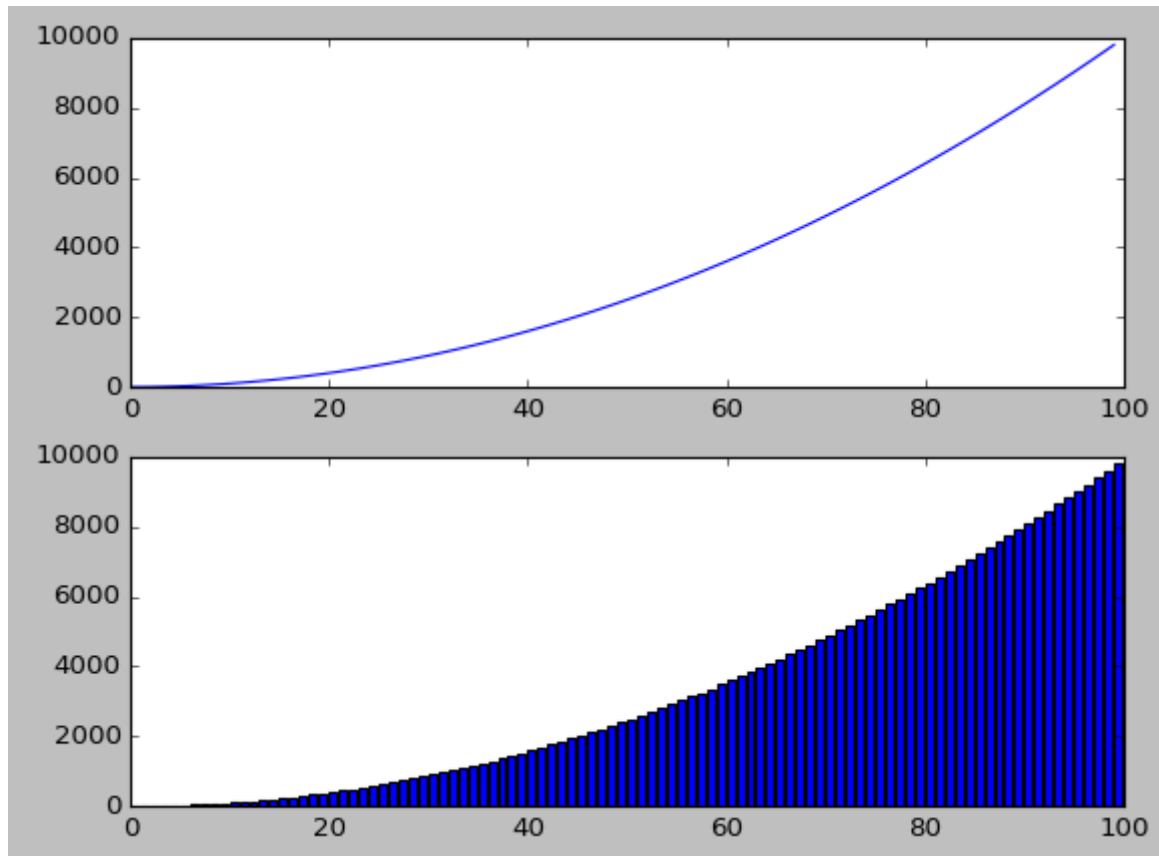
```
fig = plt.figure()
ax1 = fig.add_subplot(2,1,1)
ax2 = fig.add_subplot(2,1,2)
x=range(0,100)
y=[v*v for v in x]
```

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

(1) fig.add\_subplot(2,1,1), fig.add\_subplot(2,1,2)을 이용하여 2행 1열의 그래프를 생성했다.

(2) 1행 1열의 그래프는 plot 그래프

(3) 2행 1열의 그래프는 bar 그래프를 그렸다.



sin, cos 그래프 그려보기

```
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)
```

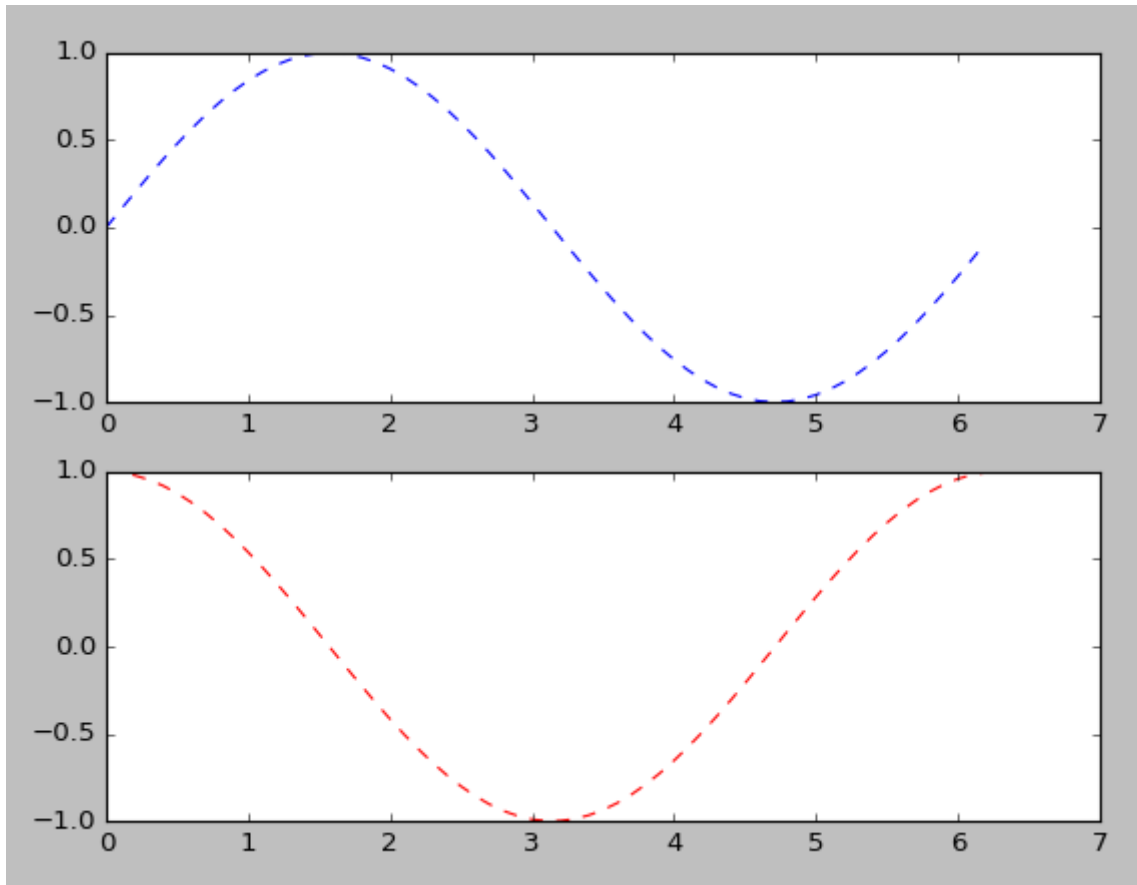


```
ax2 = fig.add_subplot(2, 1, 2)
```

```
ax1.plot(x, sin_y, 'b--')
```

```
ax2.plot(x, cos_y, 'r--')
```

```
plt.show()
```



## (실습2)

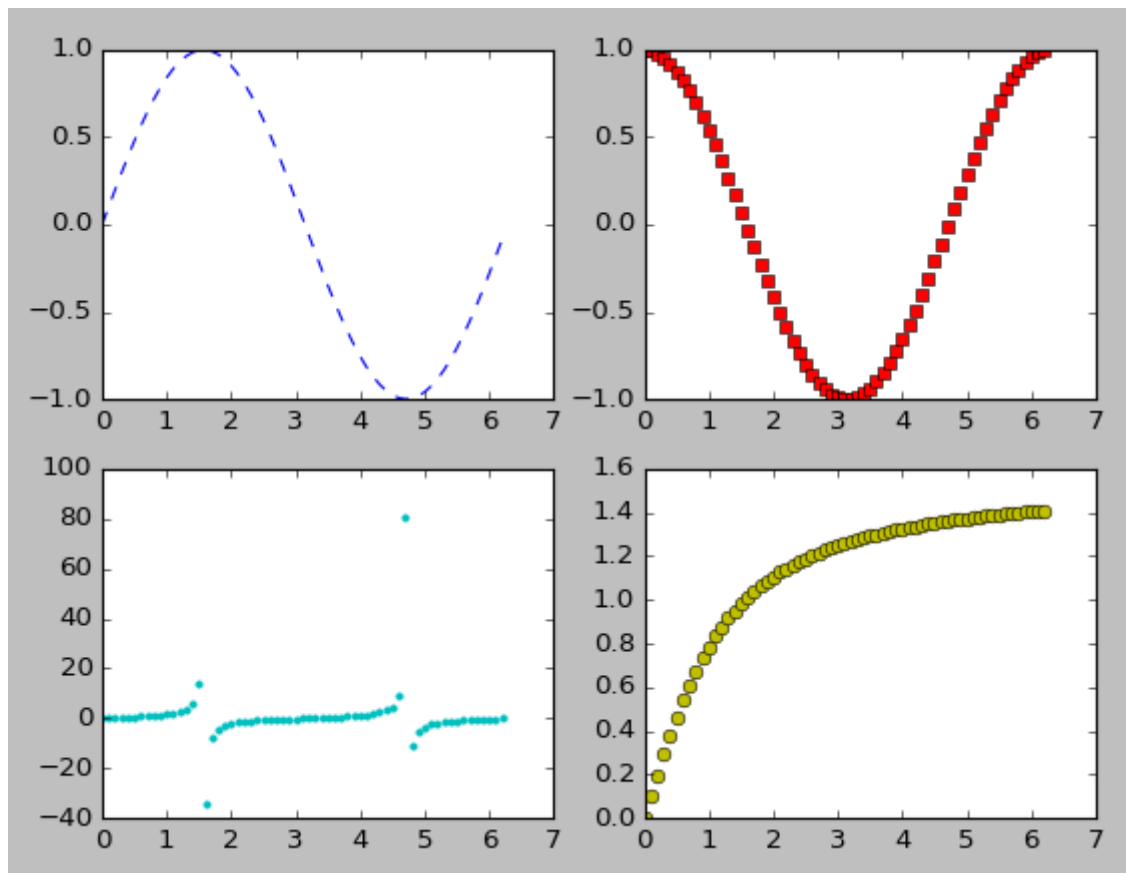
2행 2열의 그래프를 그려보자.

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 라벨 및 범례 표시하기

그래프를 그릴 때 가장 주의해야 할 것은 그래프의 x축과 y축의 값이 어떤 데이터인지 표시해 주어야 한다.

**sin, cos 그래프에 축 제목 표시**

```
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--')

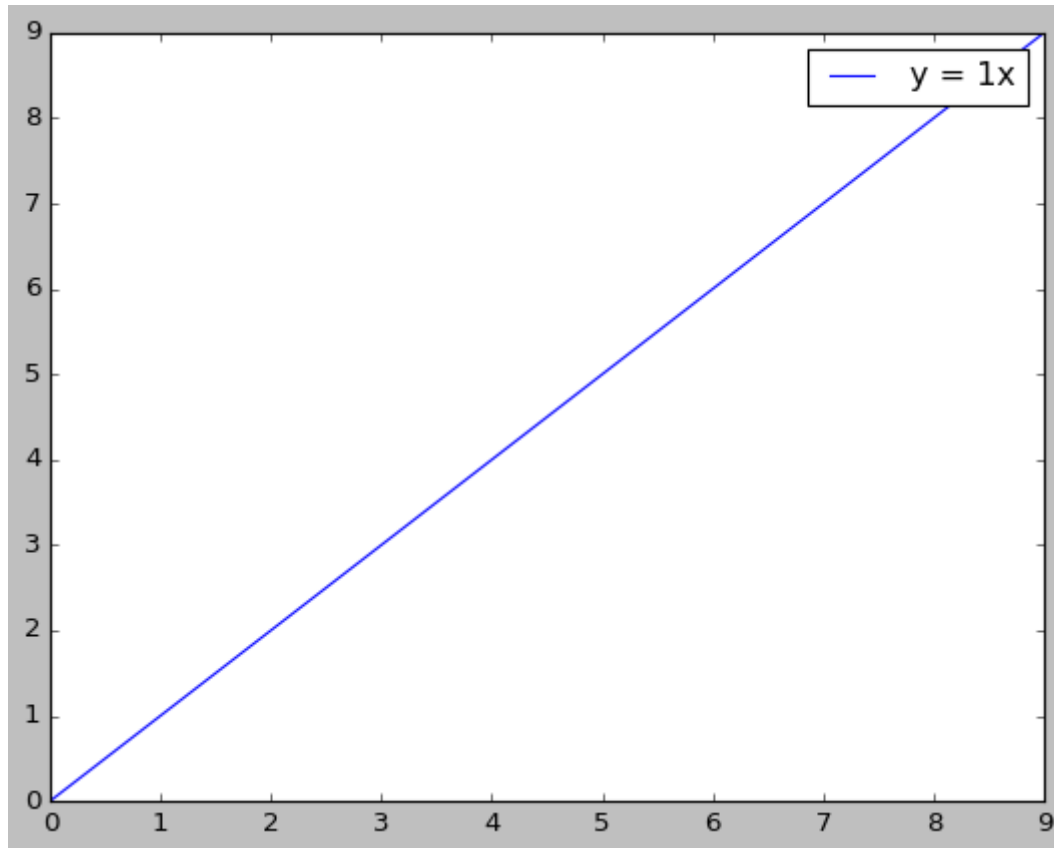
ax1.set_xlabel('x')
ax1.set_ylabel('sin(x)')

ax2.set_xlabel('x')
ax2.set_ylabel('cos(x)')

plt.show()
```

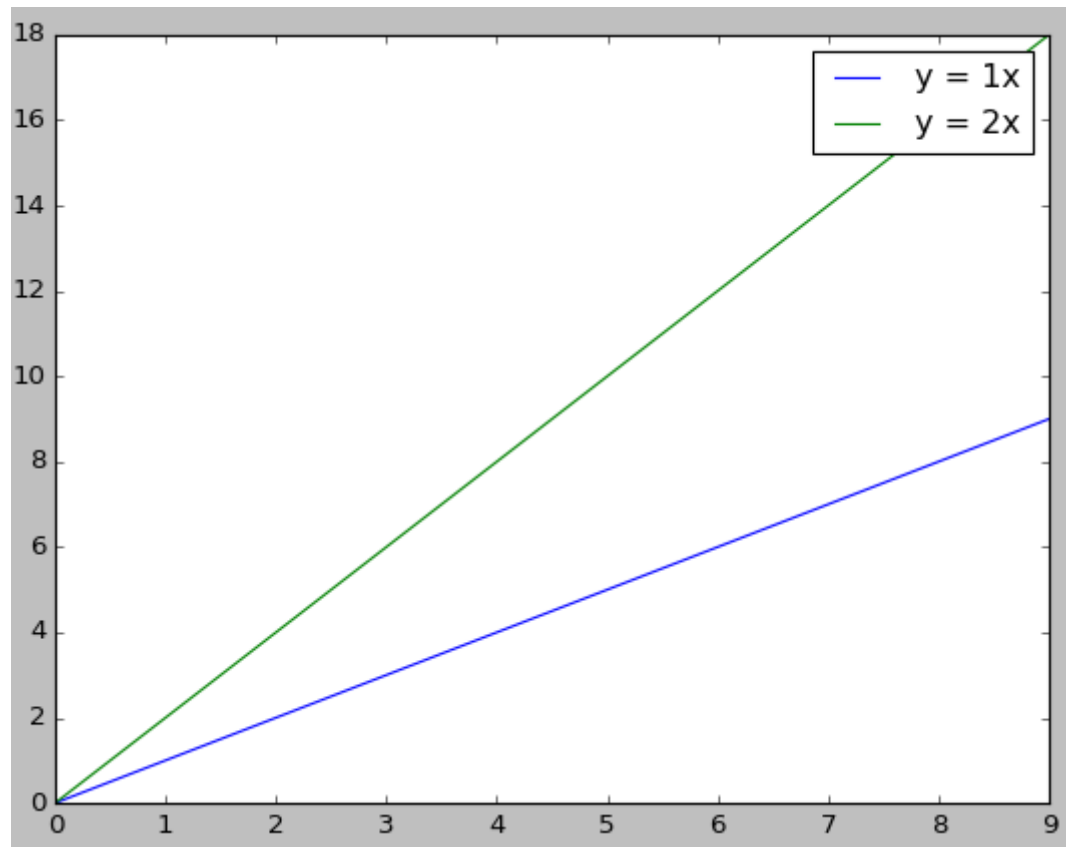
## 범례 표시 - 기본

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



### 범례 표시 - 기본(두개의 선 그래프)

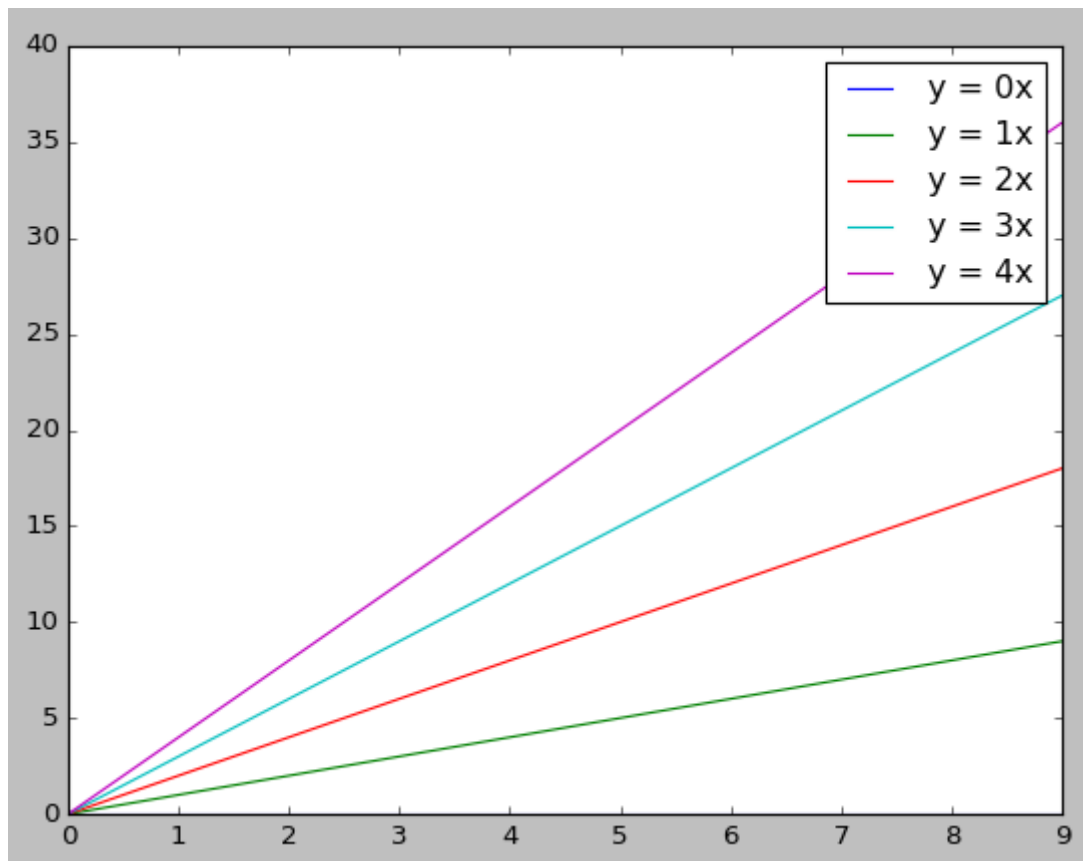
```
x = np.arange(10)
plt.plot(x, 1*x, label='y = %ix' % 1)
plt.plot(x, 2*x, label='y = %ix' % 2)
plt.legend()
plt.show()
```



### 범례 표시 - 기본(다섯개의 선 그래프)

```
x = np.arange(10)
for i in range(5):
    plt.plot(x, i * x, label='y = %ix' % i)
```

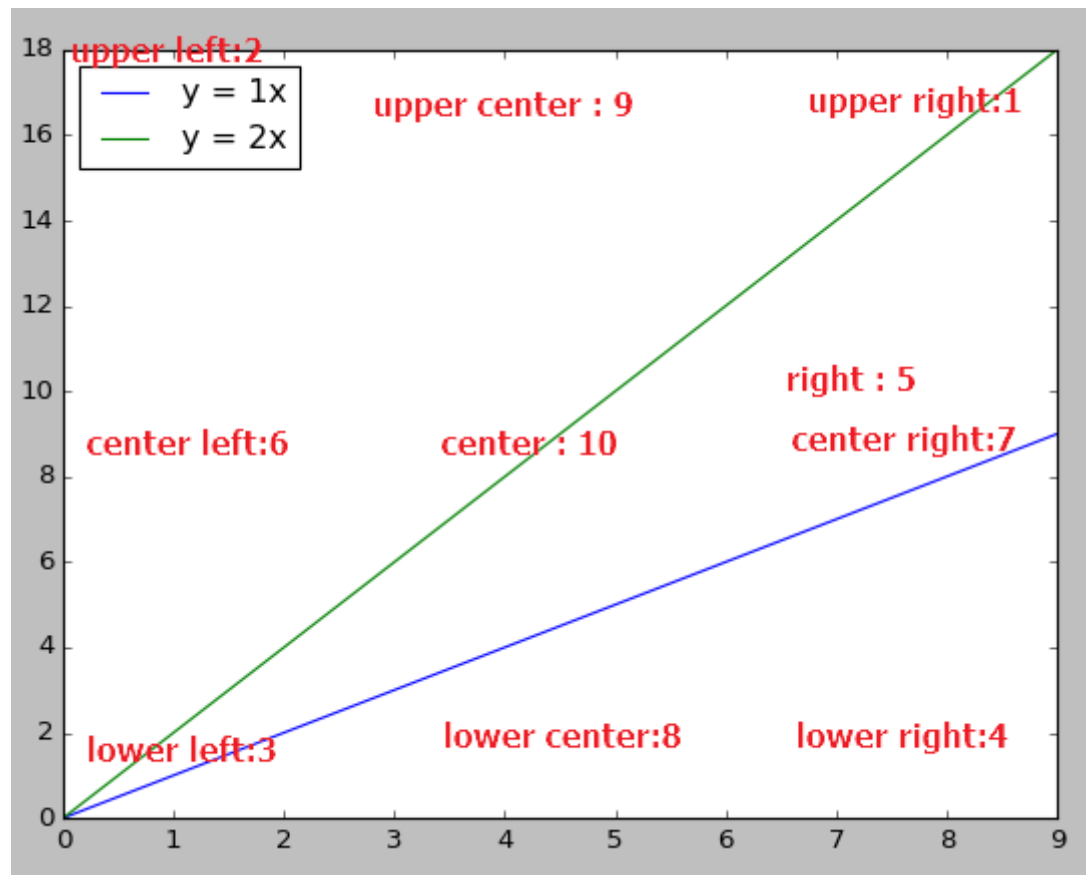
```
plt.legend()
plt.show()
```



범례의 위치를 바꿀 수 있다.

범례의 위치 변경

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



## 참고자료

### (1) Matplotlib 범례

[https://matplotlib.org/users/legend\\_guide.html](https://matplotlib.org/users/legend_guide.html)

bar 차트 그리기 :

pie 차트 그리기 : <https://wikidocs.net/4768>