

2주차 데이터시각화_과제(01분반 20171703 정태원)

In [27]:

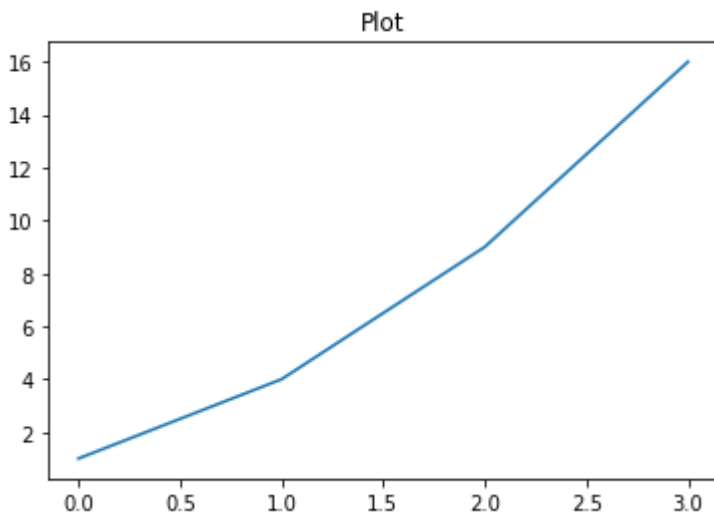
```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
```

In [4]:

```
%matplotlib inline
```

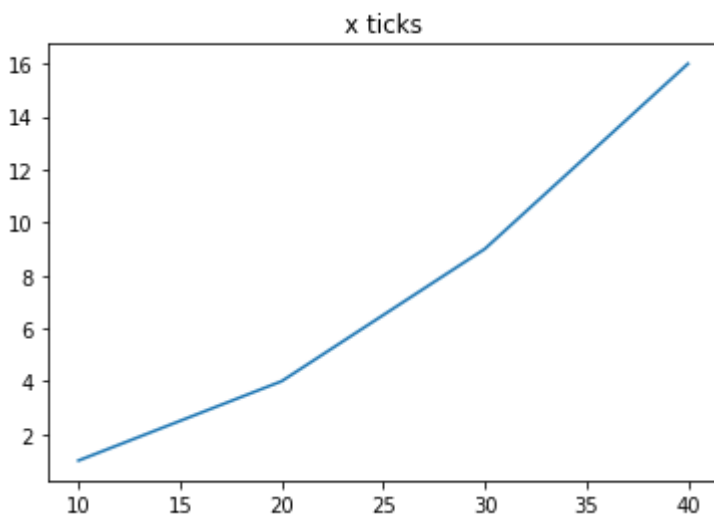
In [5]:

```
plt.title("Plot")
plt.plot([1, 4, 9, 16]) #데이터 리스트를 넘기는 중
plt.show()
```



In [6]:

```
plt.title("x ticks")
plt.plot([10, 20, 30, 40],[1, 4, 9, 16]) #x tick의 위치를 명시해주었다.
plt.show()
```



In [8]:

```
set(sorted([f.name for f in mpl.font_manager.fontManager.ttflist]))
```

Out[8]:

```
{'Abyssinica SIL',  
'Bitstream Vera Sans',  
'Bitstream Vera Sans Mono',  
'Bitstream Vera Serif',  
'DejaVu Sans',  
'DejaVu Sans Display',  
'DejaVu Sans Mono',  
'DejaVu Serif',  
'DejaVu Serif Display',  
'FreeMono',  
'FreeSans',  
'FreeSerif',  
'Garuda',  
'KacstArt',  
'KacstBook',  
'KacstDecorative',  
'KacstDigital',  
'KacstFarsi',  
'KacstLetter',  
'KacstNaskh',  
'KacstOffice',  
'KacstOne',  
'KacstPen',  
'KacstPoster',  
'KacstQurn',  
'KacstScreen',  
'KacstTitle',  
'KacstTitleL',  
'Khmer OS',  
'Khmer OS System',  
'Kinnari',  
'LKLUG',  
'Laksaman',  
'Lato',  
'Liberation Mono',  
'Liberation Sans',  
'Liberation Sans Narrow',  
'Liberation Serif',  
'Lohit Punjabi',  
'Loma',  
'NanumBarunGothic',  
'NanumGothic',  
'NanumGothicCoding',  
'NanumMyeongjo',  
'Norasi',  
'Noto Sans CJK JP',  
'OpenSymbol',  
'Padauk',  
'Padauk Book',  
'Phetsarath OT',  
'Purisa',  
'STIX',  
'STIX Math',  
'STIXGeneral',  
'STIXIntegralsD',
```

```
'STIXIntegralsSm',
'STIXIntegralsUp',
'STIXIntegralsUpD',
'STIXIntegralsUpSm',
'STIXNonUnicode',
'STIXSizeFiveSym',
'STIXSizeFourSym',
'STIXSizeOneSym',
'STIXSizeThreeSym',
'STIXSizeTwoSym',
'STIXVariants',
'Saab',
'Sawasdee',
'Symbola',
'TakaoPGothic',
'Tibetan Machine Uni',
'Tlwg Mono',
'Tlwg Typewriter',
'Tlwg Typist',
'Tlwg Typo',
'Ubuntu',
'Ubuntu Condensed',
'Ubuntu Mono',
'Umpush',
'UnBatang',
'UnDinaru',
'UnDotum',
'UnGraphic',
'UnGungseo',
'UnPilgi',
'Waree',
'cmb10',
'cmex10',
'cmmil10',
'cmr10',
'cmss10',
'cmsy10',
'cmtt10',
'esint10',
'eufm10',
'mry_KacstQurn',
'msam10',
'msbm10',
'rsfs10',
'stmary10',
'wasy10'}
```

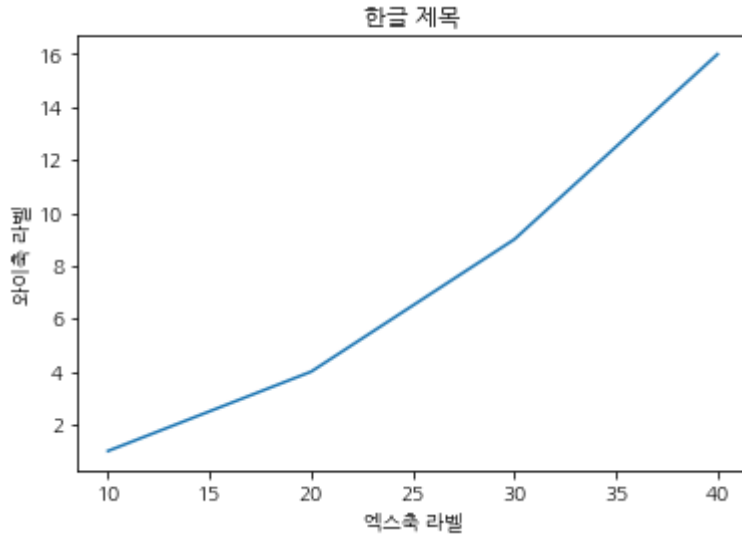
rc parameter 사용해서 그림 전체에 적용해 보기

In [10]:

```
# setting font
mpl.rc('font', family = 'NanumGothic')
# setting - in unicords
```

In [11]:

```
plt.title('한글 제목')  
plt.plot([10, 20, 30, 40],[1, 4, 9, 16])  
plt.xlabel("엑스축 라벨")  
plt.ylabel("와이축 라벨")  
plt.show()
```

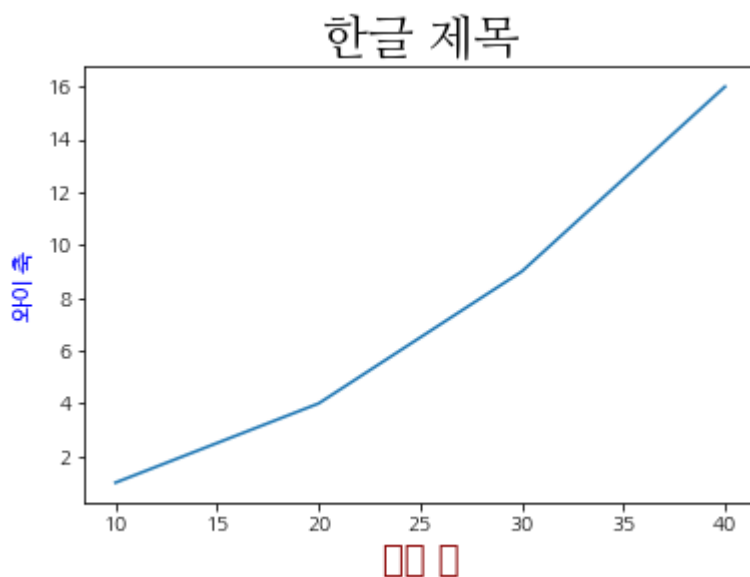


객체마다 별도의 폰트를 적용해보기

In [15]:

```
font1 = {'family': 'NanumMyeongjo', 'size': 24, 'color': 'black'}
font2 = {'family': 'NanumBarunpen', 'size': 18, 'weight': 'bold',
        'color': 'darkred'}
font3 = {'family': 'NanumBarunGothic', 'size': 12, 'weight': 'light',
        'color': 'blue'}
plt.plot([10,20,30,40], [1,4,9,16])
plt.title('한글 제목', fontdict = font1)
plt.xlabel('엑스 축', fontdict = font2)
plt.ylabel('와이 축', fontdict = font3)
plt.show()
```

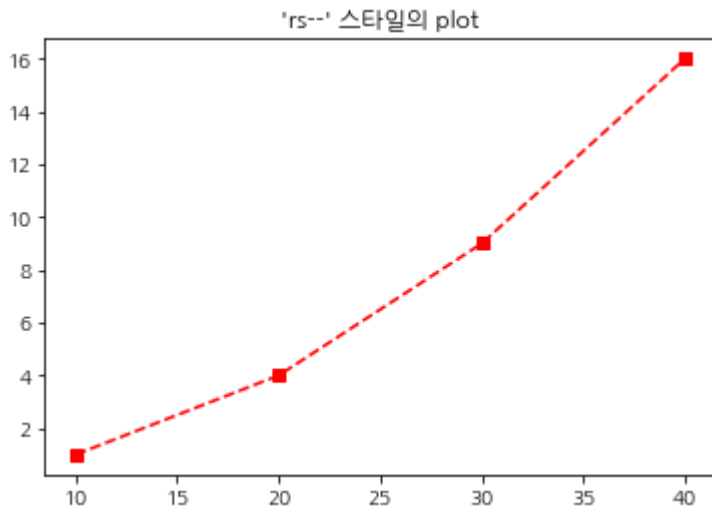
```
/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib
b/backends/backend_agg.py:211: RuntimeWarning: Glyph 50641 missing fro
m current font.
    font.set_text(s, 0.0, flags=flags)
/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib
b/backends/backend_agg.py:211: RuntimeWarning: Glyph 49828 missing fro
m current font.
    font.set_text(s, 0.0, flags=flags)
/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib
b/backends/backend_agg.py:211: RuntimeWarning: Glyph 52629 missing fro
m current font.
    font.set_text(s, 0.0, flags=flags)
/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib
b/backends/backend_agg.py:180: RuntimeWarning: Glyph 50641 missing fro
m current font.
    font.set_text(s, 0, flags=flags)
/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib
b/backends/backend_agg.py:180: RuntimeWarning: Glyph 49828 missing fro
m current font.
    font.set_text(s, 0, flags=flags)
/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib
b/backends/backend_agg.py:180: RuntimeWarning: Glyph 52629 missing fro
m current font.
    font.set_text(s, 0, flags=flags)
```



style 지정

In [16]:

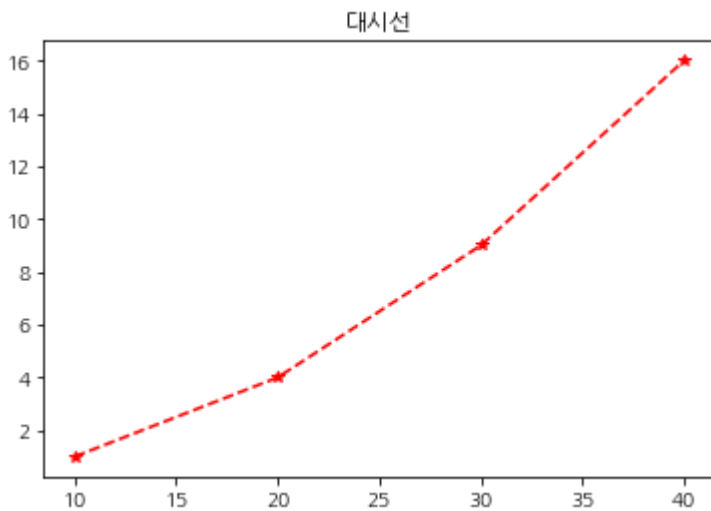
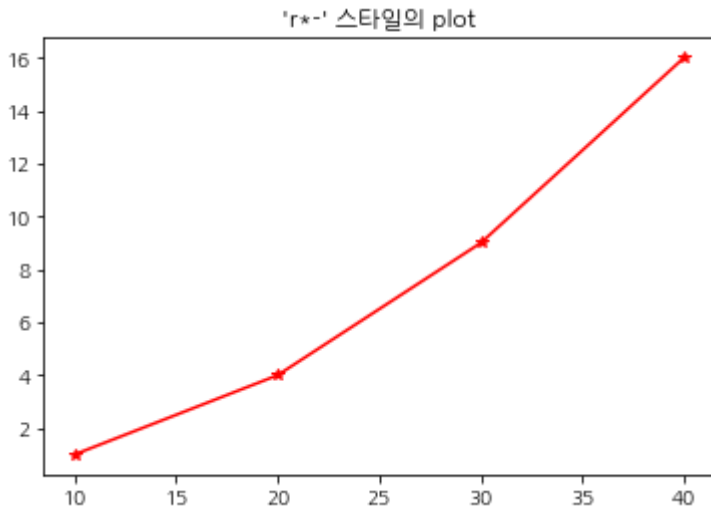
```
plt.title("'rs--' 스타일의 plot")  
plt.plot([10, 20, 30, 40], [1, 4, 9, 16], 'rs--')  
plt.show()
```

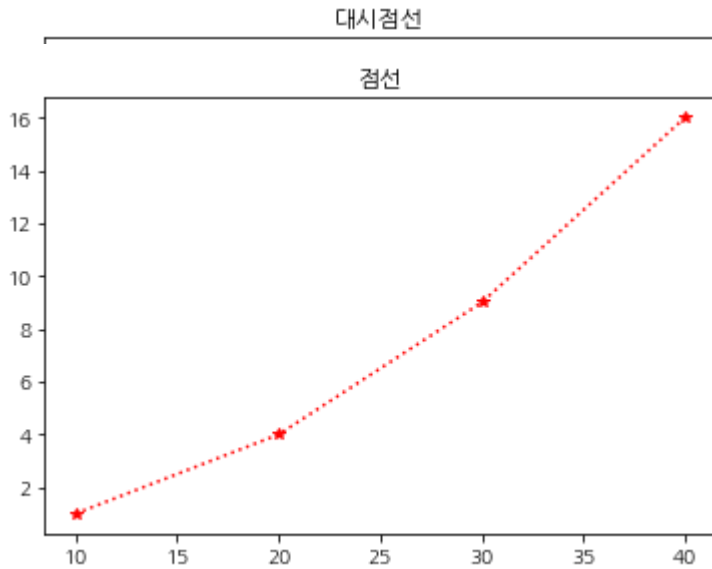


다양한 마커 사용해보기

In [18]:

```
plt.title("'r*-' 스타일의 plot")
plt.plot([10, 20, 30, 40], [1, 4, 9, 16], 'r*-' ) #스타마크
plt.show()
plt.title("대시선")
plt.plot([10, 20, 30, 40], [1, 4, 9, 16], 'r*--' ) #스타마크
plt.show()
plt.title("대시점선")
plt.plot([10, 20, 30, 40], [1, 4, 9, 16], 'r*-.') #스타마크
plt.show()
plt.title("점선")
plt.plot([10, 20, 30, 40], [1, 4, 9, 16], 'r*:') #스타마크
plt.show()
```





Matplotlib의 style 인수 목록

In [21]:

```
plt.plot([10, 20, 30, 40], [1, 4, 9, 16], c="b", lw=5, ls="--", marker="o", ms=15)
plt.title("스타일 적용의 예")
plt.show()
```

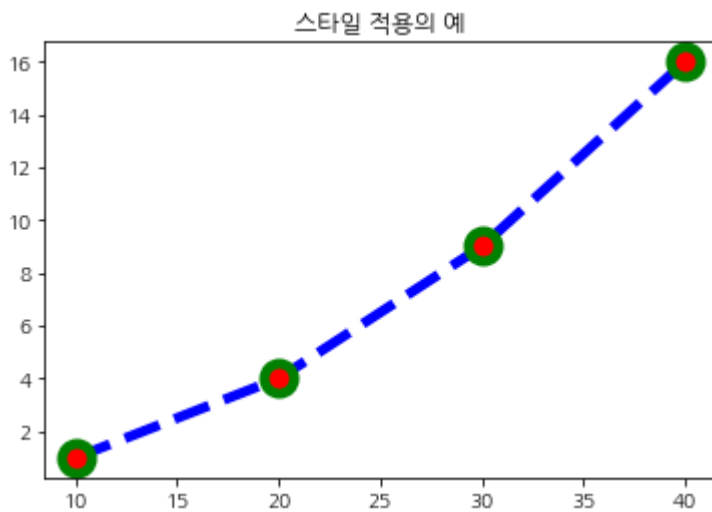


그림 범위 지정

In [24]:

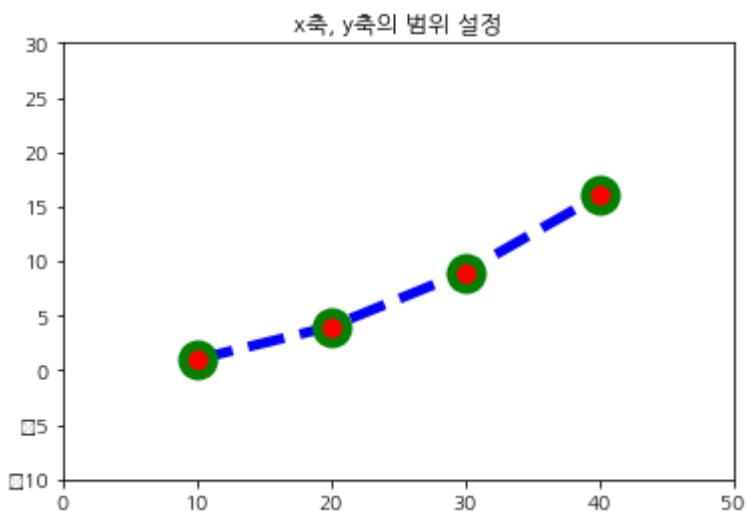
```
plt.title("x축, y축의 범위 설정")
plt.plot([10, 20, 30, 40], [1, 4, 9, 16],
         c="b", lw=5, ls="--", marker="o", ms=15, mec="g", mew=5, mfc="r")
plt.xlim(0, 50)
plt.ylim(-10, 30)
plt.show()
```

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:211: RuntimeWarning: Glyph 8722 missing from current font.

```
font.set_text(s, 0.0, flags=flags)
```

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:180: RuntimeWarning: Glyph 8722 missing from current font.

```
font.set_text(s, 0, flags=flags)
```



In [33]:

```

X = np.linspace(-np.pi, np.pi, 256)
C = np.cos(X)
plt.title("x축과 y축의 tick label 설정")
plt.plot(X, C)
plt.xticks([-np.pi, -np.pi / 2, np.pi / 2, np.pi])
plt.yticks([-1, 0,
            +1])
plt.show()

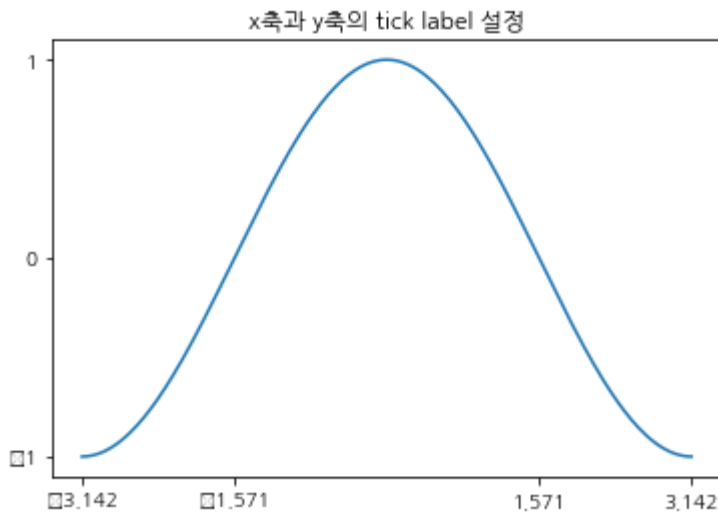
```

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:211: RuntimeWarning: Glyph 8722 missing from current font.

font.set_text(s, 0.0, flags=flags)

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:180: RuntimeWarning: Glyph 8722 missing from current font.

font.set_text(s, 0, flags=flags)



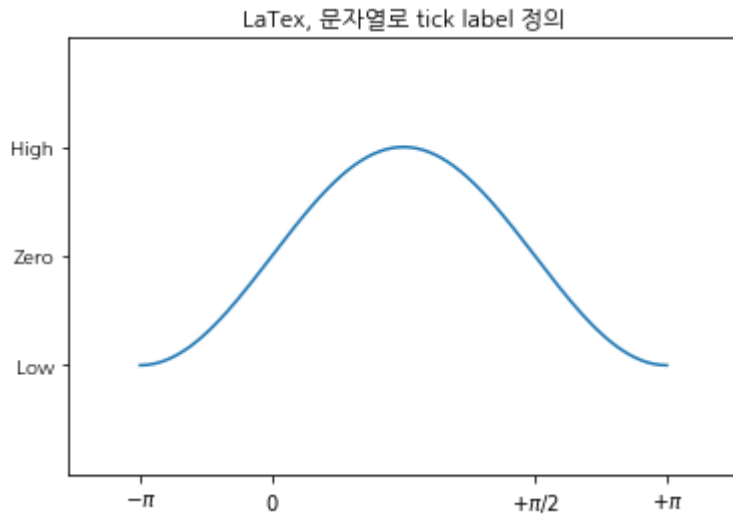
Latex로 작성하기

In [40]:

```

X = np.linspace(-np.pi, np.pi, 256)
C = np.cos(X)
plt.title("LaTeX, 문자열로 tick label 정의")
plt.plot(X, C)
plt.xticks([-np.pi, -np.pi / 2.0, np.pi / 2, np.pi], [r'$-\pi$', r'$0$', r'$+\pi/2$'])
plt.yticks([-1, 0, 1], ["Low", "Zero", "High"])
plt.xlim(-4, 4)
plt.ylim(-2, 2)
plt.show()

```

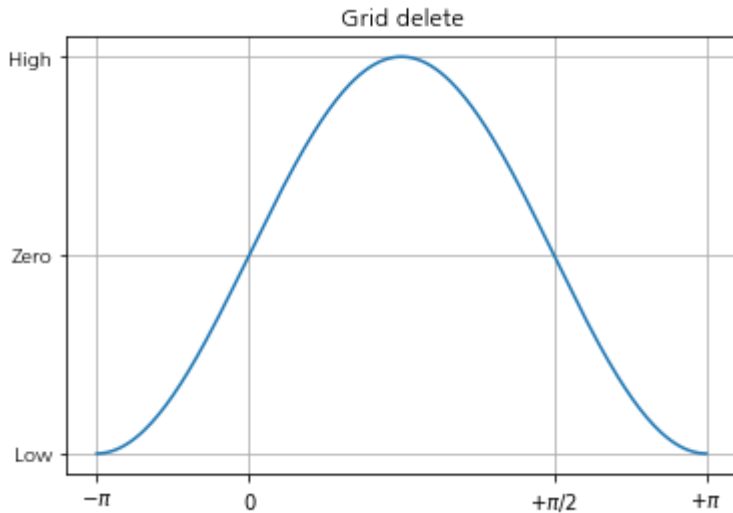


Grid setting

In [48]:

```
X = np.linspace(-np.pi, np.pi, 256)
C = np.cos(X)
plt.plot(X, C)
plt.title("Grid delete")

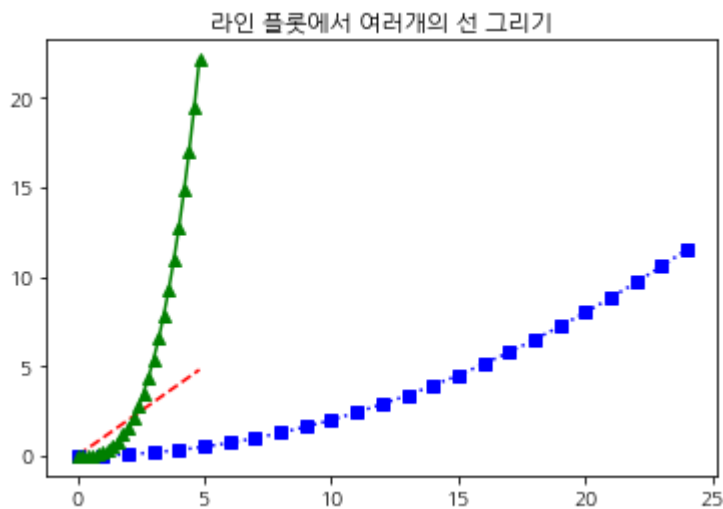
plt.xticks([-np.pi, -np.pi/2, np.pi/2, np.pi], [r'$-\pi$', r'$0$', r'$+\pi/2$', r'$+\pi$'])
plt.yticks([-1, 0, 1], ["Low", "Zero", "High"])
plt.grid(True)
plt.show()
```



여러개의 선을 그리기

In [56]:

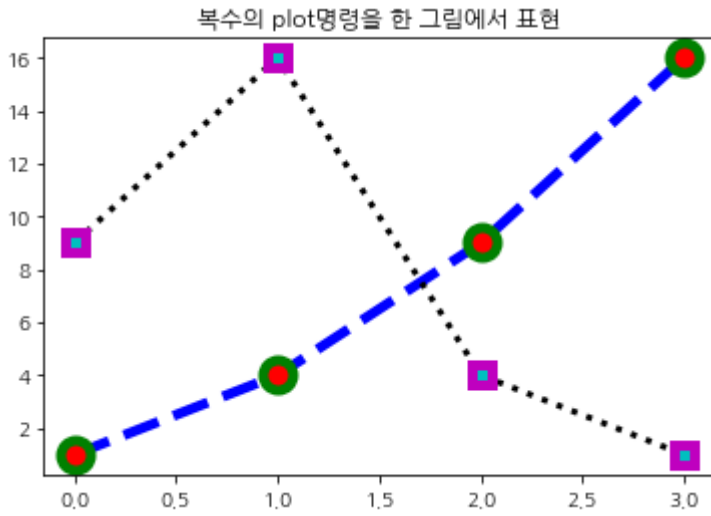
```
t = np.arange(0., 5., 0.2)
plt.title("라인 플롯에서 여러개의 선 그리기")
plt.plot(t, t, 'r--', 0.5*t**2, 'bs:', t, 0.2*t**3, 'g^-')
plt.show()
```



겹쳐그리기

In [55]:

```
plt.title("복수의 plot명령을 한 그림에서 표현")
plt.plot([1,4,9, 16],c="b", lw=5, ls="--", marker = "o", ms= 15, mec = "g",mew = 5,
plt.plot([9,16,4,1], c= "k",lw=3, ls=":", marker= "s", ms = 10, mec= "m", mew = 5 ,
plt.show()
```



범례(Legend)

In [60]:

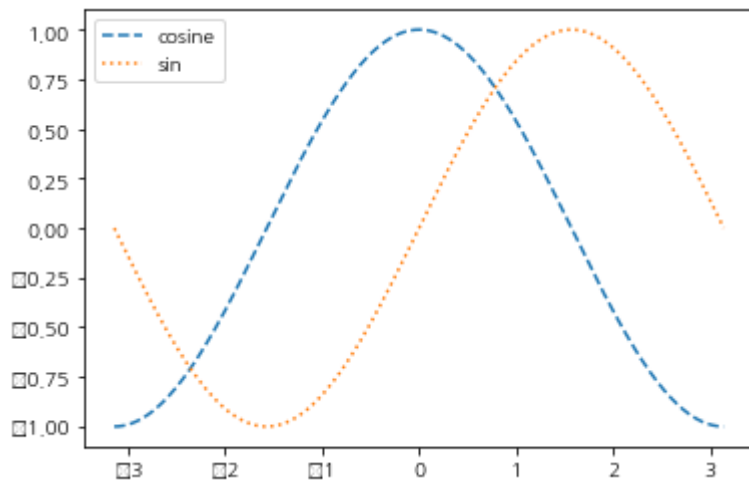
```
X = np.linspace(-np.pi, np.pi, 256)
C, S = np.cos(X), np.sin(X)
plt.plot(X, C, ls="--", label="cosine")
plt.plot(X, S, ls=":", label="sin")
plt.legend(loc=2)
plt.show()
```

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:211: RuntimeWarning: Glyph 8722 missing from current font.

```
font.set_text(s, 0.0, flags=flags)
```

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:180: RuntimeWarning: Glyph 8722 missing from current font.

```
font.set_text(s, 0, flags=flags)
```



x축, y축 라벨, 타이틀

In [63]:

```
X = np.linspace(-np.pi, np.pi, 256)
C,S = np.cos(X), np.sin(X)
plt.plot(X, C, label = "cosine")
plt.xlabel("time")
plt.ylabel("amplitude")
plt.title("Cosine Plot")
plt.show()
```

/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:211: RuntimeWarning: Glyph 8722 missing from current font.

font.set_text(s, 0.0, flags=flags)

/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:180: RuntimeWarning: Glyph 8722 missing from current font.

font.set_text(s, 0, flags=flags)

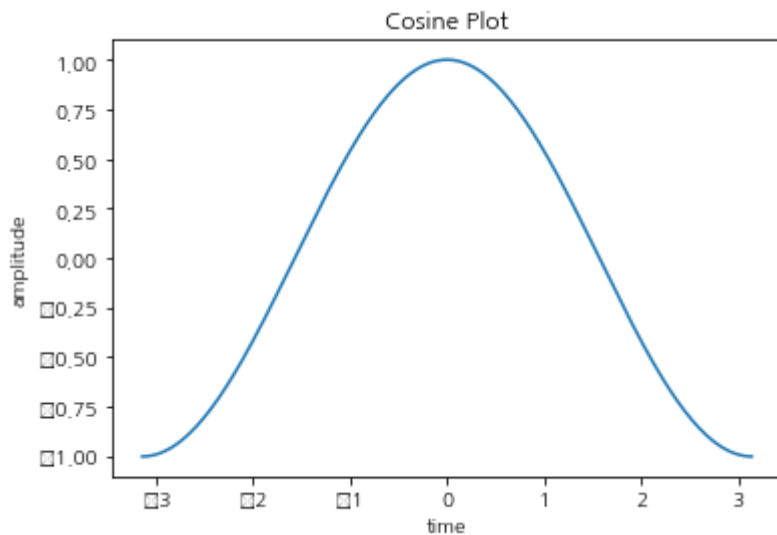


Figure 객체

In [64]:

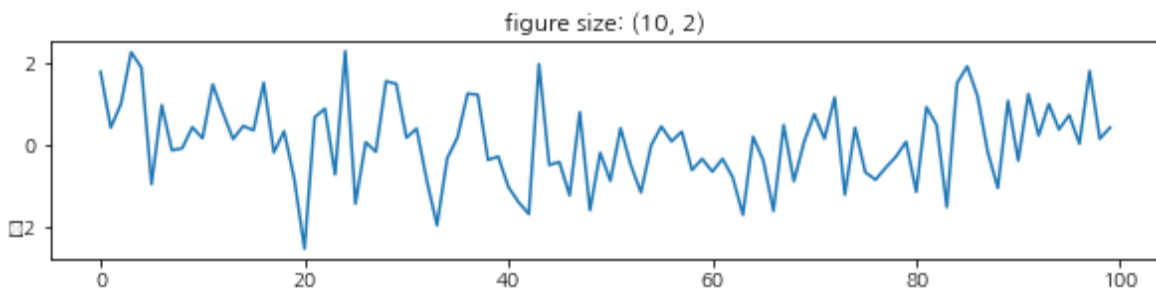
```
np.random.seed(0)
f1= plt.figure(figsize=(10, 2))
plt.title("figure size: (10, 2)")
plt.plot(np.random.randn(100))
plt.show()
```

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:211: RuntimeWarning: Glyph 8722 missing from current font.

```
font.set_text(s, 0.0, flags=flags)
```

/home/studyingession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:180: RuntimeWarning: Glyph 8722 missing from current font.

```
font.set_text(s, 0, flags=flags)
```



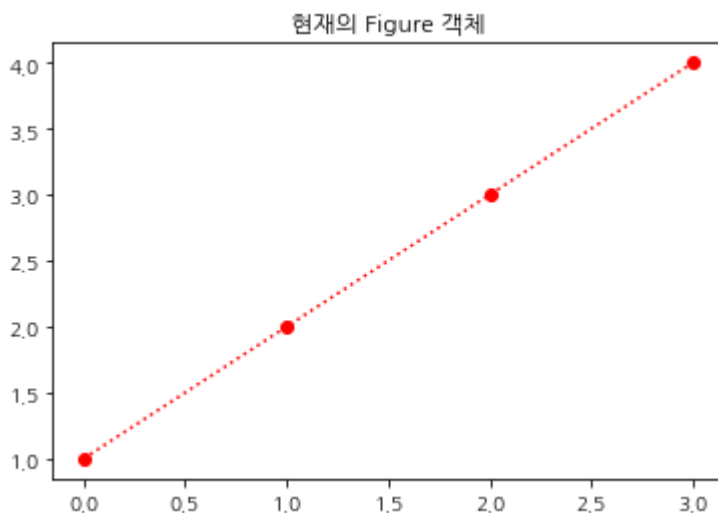
In [65]:

```
f1 = plt.figure(1)
plt.title("현재의 Figure 객체")
plt.plot([1,2,3,4], 'ro:')

f2 = plt.gcf()
print(f1, id(f1))
print(f2, id(f2))
plt.show()
```

Figure(432x288) 140536211462800

Figure(432x288) 140536211462800



In [67]:

```

x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)
y1 = np.cos(2* np.pi*x1)*np.exp(-x1)
y2 = np.cos(2*np.pi*x2)

ax1 = plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'yo-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')
print(ax1)

ax2 = plt.subplot(2, 1, 2)
plt.plot(x2, y2, 'r.-')
plt.xlabel('Undamped')
print(ax2)

plt.tight_layout()
plt.show()

```

AxesSubplot(0.125,0.536818;0.775x0.343182)

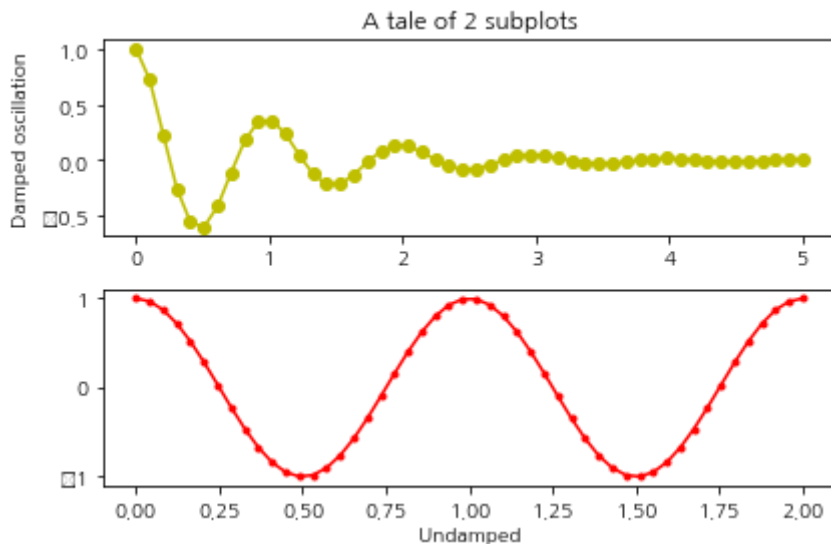
AxesSubplot(0.125,0.125;0.775x0.343182)

/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:211: RuntimeWarning: Glyph 8722 missing from current font.

font.set_text(s, 0.0, flags=flags)

/home/studyingssession/anaconda3/lib/python3.7/site-packages/matplotlib/backends/backend_agg.py:180: RuntimeWarning: Glyph 8722 missing from current font.

font.set_text(s, 0, flags=flags)



In [68]:

```
np.random.seed(0)

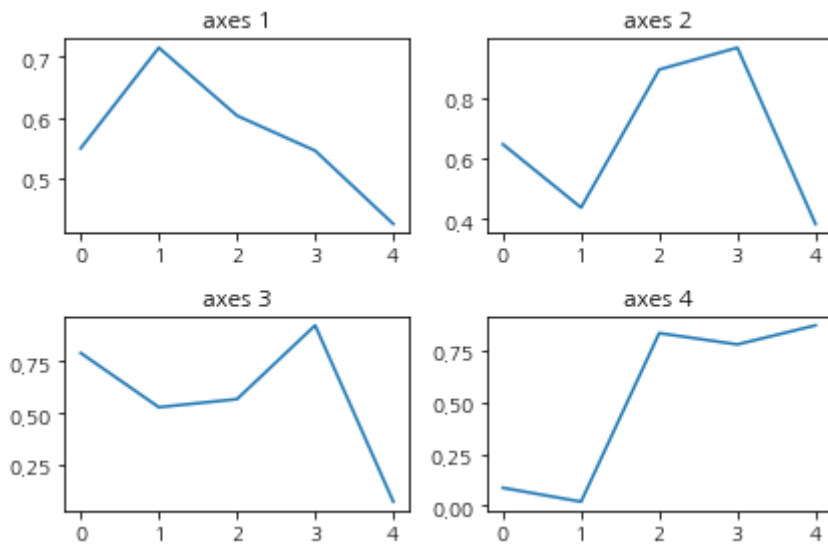
plt.subplot(221)
plt.plot(np.random.rand(5))
plt.title("axes 1")

plt.subplot(222)
plt.plot(np.random.rand(5))
plt.title("axes 2")

plt.subplot(223)
plt.plot(np.random.rand(5))
plt.title("axes 3")

plt.subplot(224)
plt.plot(np.random.rand(5))
plt.title("axes 4")

plt.tight_layout()
plt.show()
```



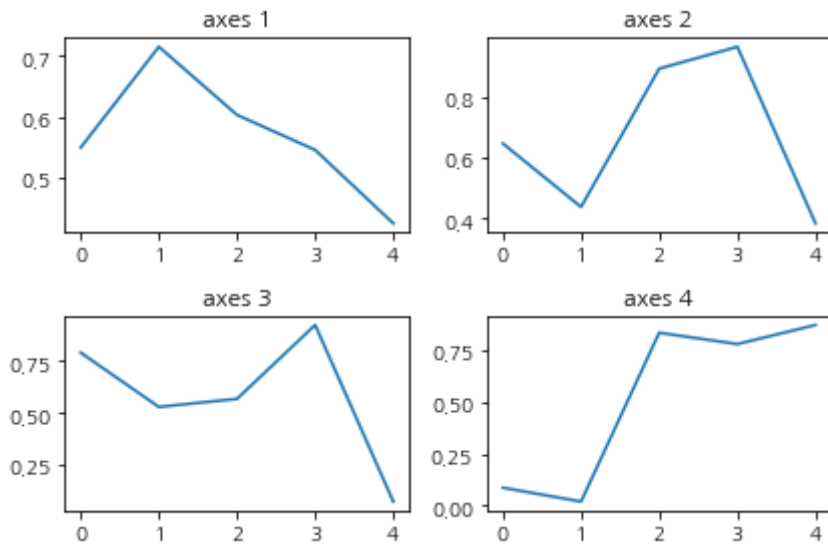
subplots

In [73]:

```
fig, axes = plt.subplots(2, 2)

np.random.seed(0)
axes[0,0].plot(np.random.rand(5))
axes[0,0].set_title("axes 1")
axes[0,1].plot(np.random.rand(5))
axes[0,1].set_title("axes 2")
axes[1,0].plot(np.random.rand(5))
axes[1,0].set_title("axes 3")
axes[1,1].plot(np.random.rand(5))
axes[1,1].set_title("axes 4")

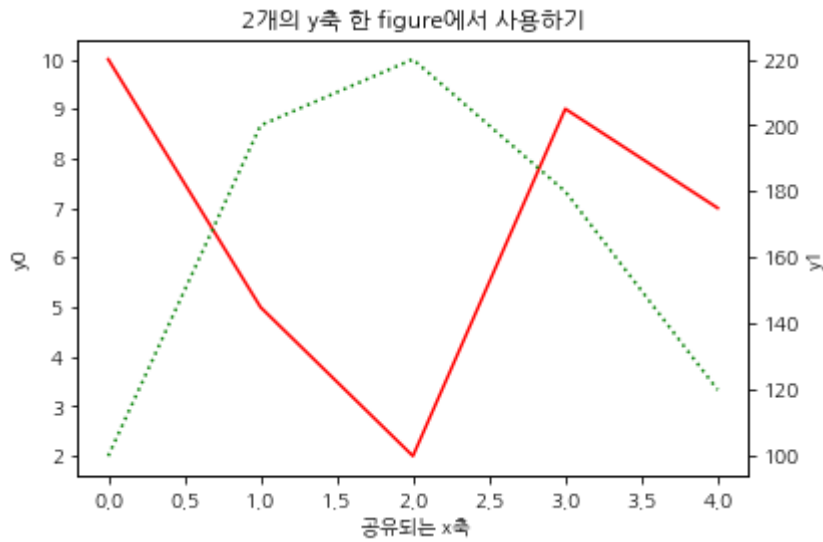
plt.tight_layout()
plt.show()
```



Axis 객체와 축

In [76]:

```
fig, ax0 = plt.subplots()
ax1 = ax0.twinx()
ax0.set_title("2개의 y축 한 figure에서 사용하기")
ax0.plot([10, 5, 2, 9, 7], 'r-', label = "y0")
ax0.set_ylabel("y0")
ax1.plot([100, 200, 220, 180, 120],
        'g:', label= "y1")
ax1.set_ylabel("y1")
ax1.grid(False)
ax0.set_xlabel("공유되는 x축")
plt.show()
```



연습 문제1 1

xlabel, ylabel, title을 모두 갖추고 있어야 한다. 하나의 Figure(일단, 그림이라고 이해한다. 아래에 자세한 설명이 있다.)에 3개 이상의 Plot을 그린다. 각 Plot은 다른 선, 마크, 색 스타일을 가진다. legend는 그래프와 겹치지 않는 곳에 위치 시키도록 한다.

In [130]:

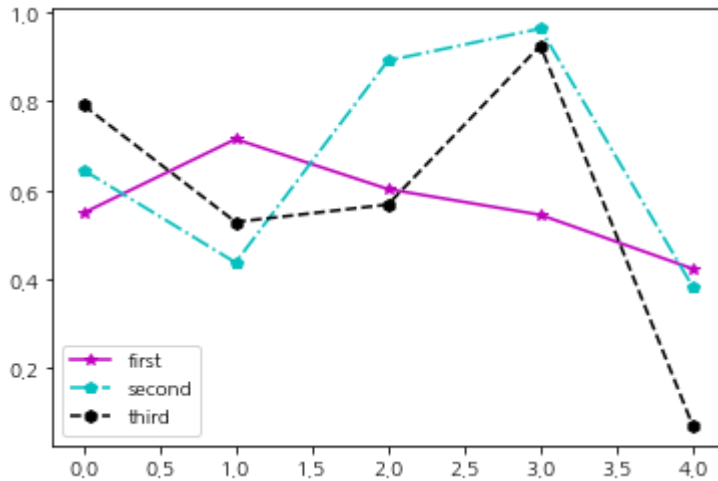
```

np.random.seed(0)
mpl.rc('font', family = 'NanumGothic')
plt.plot(np.random.rand(5), 'm*-', label = 'first')
plt.plot(np.random.rand(5), 'cp-.' , label= 'second')
plt.plot(np.random.rand(5), 'kh--', label= 'third')
plt.legend(loc=3)

```

Out[130]:

<matplotlib.legend.Legend at 0x7fd11ffd9290>



연습 문제2

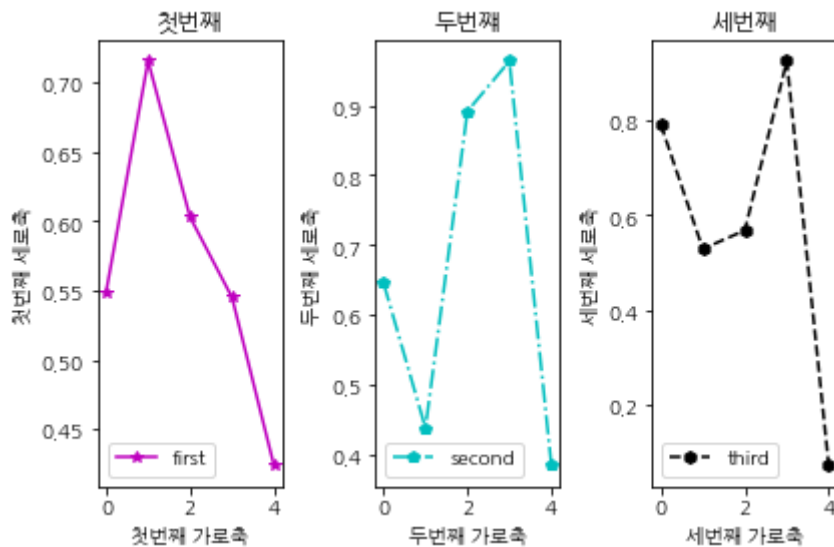
여러가지 함수를 사용하여 위와 같이 subplot들로 구성된 그림을 그려보자. 모든 subplot에 대해 xlabel, ylabel, title이 있어야 한다.

In [128]:

```

np.random.seed(0)
mpl.rc('font', family = 'NanumGothic')
plt.subplot(131)
plt.plot(np.random.rand(5), 'm*-', label = 'first')
plt.xlabel("첫번째 가로축")
plt.ylabel("첫번째 세로축")
plt.title("첫번째")
plt.legend(loc=3)
plt.subplot(132)
plt.plot(np.random.rand(5), 'cp-.', label= 'second')
plt.xlabel("두번째 가로축")
plt.ylabel("두번째 세로축")
plt.title("두번째")
plt.legend(loc=3)
plt.subplot(133)
plt.plot(np.random.rand(5), 'kh--', label= 'third')
plt.xlabel("세번째 가로축")
plt.ylabel("세번째 세로축")
plt.title("세번째")
plt.legend(loc=3)
plt.tight_layout()
plt.show()

```



연습문제3

Matplotlib 갤러리 웹사이트에서 관심있는 예제 코드를 하나 고른다. <http://matplotlib.org/gallery.html> (<http://matplotlib.org/gallery.html>) 예제 코드에 사용된 Matplotlib API 명령의 목록을 만들고 Matplotlib 웹사이트에서 관련 링크를 찾아 내용을 정리한다. 변형된 형태의 플롯을 만들어본다.

In [11]:

#선택예제

```

import numpy as np
import matplotlib.pyplot as plt

def plot_scatter(ax, prng, nb_samples=100):
    """Scatter plot.
    """
    for mu, sigma, marker in [(-.5, 0.75, 'o'), (0.75, 1., 's')]:
        x, y = prng.normal(loc=mu, scale=sigma, size=(2, nb_samples))
        ax.plot(x, y, ls='none', marker=marker)
    ax.set_xlabel('X-label')
    return ax

def plot_colored_sinusoidal_lines(ax):
    """Plot sinusoidal lines with colors following the style color cycle.
    """
    L = 2 * np.pi
    x = np.linspace(0, L)
    nb_colors = len(plt.rcParams['axes.prop_cycle'])
    shift = np.linspace(0, L, nb_colors, endpoint=False)
    for s in shift:
        ax.plot(x, np.sin(x + s), '-')
    ax.set_xlim([x[0], x[-1]])
    return ax

def plot_bar_graphs(ax, prng, min_value=5, max_value=25, nb_samples=5):
    """Plot two bar graphs side by side, with letters as x-tick labels.
    """
    x = np.arange(nb_samples)
    ya, yb = prng.randint(min_value, max_value, size=(2, nb_samples))
    width = 0.25
    ax.bar(x, ya, width)
    ax.bar(x + width, yb, width, color='C2')
    ax.set_xticks(x + width)
    ax.set_xticklabels(['a', 'b', 'c', 'd', 'e'])
    return ax

def plot_colored_circles(ax, prng, nb_samples=15):
    """Plot circle patches.

    NB: draws a fixed amount of samples, rather than using the length of
    the color cycle, because different styles may have different numbers
    of colors.
    """
    for sty_dict, j in zip(plt.rcParams['axes.prop_cycle'], range(nb_samples)):
        ax.add_patch(plt.Circle(prng.normal(scale=3, size=2),
                                radius=1.0, color=sty_dict['color']))
    # Force the limits to be the same across the styles (because different
    # styles may have different numbers of available colors).
    ax.set_xlim([-4, 8])
    ax.set_ylim([-5, 6])
    ax.set_aspect('equal', adjustable='box') # to plot circles as circles
    return ax

```

```

def plot_image_and_patch(ax, prng, size=(20, 20)):
    """Plot an image with random values and superimpose a circular patch.
    """
    values = prng.random_sample(size=size)
    ax.imshow(values, interpolation='none')
    c = plt.Circle((5, 5), radius=5, label='patch')
    ax.add_patch(c)
    # Remove ticks
    ax.set_xticks([])
    ax.set_yticks([])

def plot_histograms(ax, prng, nb_samples=10000):
    """Plot 4 histograms and a text annotation.
    """
    params = ((10, 10), (4, 12), (50, 12), (6, 55))
    for a, b in params:
        values = prng.beta(a, b, size=nb_samples)
        ax.hist(values, histtype="stepfilled", bins=30, alpha=0.8, normed=True)
    # Add a small annotation.
    ax.annotate('Annotation', xy=(0.25, 4.25), xycoords='data',
                xytext=(0.9, 0.9), textcoords='axes fraction',
                va="top", ha="right",
                bbox=dict(boxstyle="round", alpha=0.2),
                arrowprops=dict(
                    arrowstyle="->",
                    connectionstyle="angle,angleA=-95,angleB=35,rad=10"),
                )
    return ax

def plot_figure(style_label=""):
    """Setup and plot the demonstration figure with a given style.
    """
    # Use a dedicated RandomState instance to draw the same "random" values
    # across the different figures.
    prng = np.random.RandomState(96917002)

    # Tweak the figure size to be better suited for a row of numerous plots:
    # double the width and halve the height. NB: use relative changes because
    # some styles may have a figure size different from the default one.
    (fig_width, fig_height) = plt.rcParams['figure.figsize']
    fig_size = [fig_width * 2, fig_height / 2]

    fig, axes = plt.subplots(ncols=6, nrows=1, num=style_label,
                             figsize=fig_size, squeeze=True)
    axes[0].set_ylabel(style_label)

    plot_scatter(axes[0], prng)
    plot_image_and_patch(axes[1], prng)
    plot_bar_graphs(axes[2], prng)
    plot_colored_circles(axes[3], prng)
    plot_colored_sinusoidal_lines(axes[4])
    plot_histograms(axes[5], prng)

    fig.tight_layout()

    return fig

```



```

if __name__ == "__main__":

    # Setup a list of all available styles, in alphabetical order but
    # the `default` and `classic` ones, which will be forced resp. in
    # first and second position.
    style_list = list(plt.style.available) # *new* list: avoids side effects.
    style_list.remove('classic') # `classic` is in the list: first remove it.
    style_list.sort()
    style_list.insert(0, u'default')
    style_list.insert(1, u'classic')

    # Plot a demonstration figure for every available style sheet.
    for style_label in style_list:
        with plt.style.context(style_label):
            fig = plot_figure(style_label=style_label)

plt.show()

```

```

/home/studyingssession/anaconda3/lib/python3.7/site-packages/ipykerne
l_launcher.py:88: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be remo
ved in 3.1. Use 'density' instead.
/home/studyingssession/anaconda3/lib/python3.7/site-packages/ipykerne
l_launcher.py:88: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be remo
ved in 3.1. Use 'density' instead.
/home/studyingssession/anaconda3/lib/python3.7/site-packages/ipykerne
l_launcher.py:88: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be remo
ved in 3.1. Use 'density' instead.
/home/studyingssession/anaconda3/lib/python3.7/site-packages/ipykerne
l_launcher.py:88: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be remo
ved in 3.1. Use 'density' instead.

```

Matplotlib API 명령의 목록

plt.rcParams['axes.prop_cycle'] - 차트의 크기를 결정 plt.Circle((5, 5), radius=5, label='pabtch') - 도형을 생성
 ax.add_patch(plt.Circle(prng.normal(scale=3, size=2), radius=1.0, color=sty_dict['color']))
 plt.style.context(style_label): circle함수를 통해 만든 원을 plot에 삽입 (fig_width, fig_height) =
 plt.rcParams['figure.figsize'] rcParams를 통해 차트 객체의 크기를 결정

plt.style.available - 가능한 스타일들 ['bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'grayscale',
 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark-palette', 'seaborn-dark', 'seaborn-darkgrid', 'seaborn-deep',
 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk',
 'seaborn-ticks', 'seaborn-white', 'seaborn-whitegrid', 'seaborn', 'Solarize_Light2', 'tableau-colorblind10',
 '_classic_test'] plt.scatter() - scatter 객체의 모양과 동작을 제어함

#아래는 numpy 함수들 prng=np.random.RandomState(96917002) plot_scatter(axes[0], prng)
 plot_image_and_patch(axes[1], prng) plot_bar_graphs(axes[2], prng) plot_colored_circles(axes[3], prng)
 plot_colored_sinusoidal_lines(axes[4]) plot_histograms(axes[5], prng)

In [48]:

```

prng=np.random.RandomState(96917002)
mu, sigma = 0, 0.1 # mean and standard deviation
s = np.random.normal(mu, sigma, 1000)
print(s)

```

```

-1.76833843e-01  3.33481793e-02  8.14319822e-02  3.89233892e-03
-1.85053671e-02 -8.07648488e-02 -1.44653470e-01  8.00297949e-02
-3.09114445e-02 -2.33466662e-02  1.73272119e-01  6.84501107e-02
 3.70825001e-02  1.42061805e-02  1.51999486e-01  1.71958931e-01
 9.29505111e-02  5.82224591e-02 -2.09460307e-01  1.23721914e-02
-1.30106954e-02  9.39532294e-03  9.43046087e-02 -2.73967717e-01
-5.69312053e-02  2.69904355e-02 -4.66845546e-02 -1.41690611e-01
 8.68963487e-02  2.76871906e-02 -9.71104570e-02  3.14817205e-02
 8.21585712e-02  5.29264630e-04  8.00564803e-02  7.82601752e-03
-3.95228983e-02 -1.15942052e-01 -8.59307670e-03  1.94292938e-02
 8.75832762e-02 -1.15107468e-02  4.57415606e-02 -9.64612014e-02
-7.82629156e-02 -1.10389299e-02 -1.05462846e-01  8.20247837e-02
 4.63130329e-02  2.79095764e-02  3.38904125e-02  2.02104356e-01
-4.68864188e-02 -2.20144129e-01  1.99300197e-02 -5.06035410e-03
-5.17519043e-02 -9.78829859e-02 -4.39189522e-02  1.81338429e-02
-5.02816701e-02  2.41245368e-01 -9.60504382e-02 -7.93117363e-02
-2.28862004e-01  2.51484415e-02 -2.01640663e-01 -5.39454633e-02
-2.75670535e-02 -7.09727966e-02  1.73887268e-01  9.94394391e-02
 1.31913688e-01 -8.82418819e-02  1.12859406e-01  4.96000946e-02
 7.71405949e-02  1.02943883e-01 -9.08763246e-02 -4.24317621e-02

```

In [47]:

#변경한 과제

np.random.seed(0)

plt.subplot(121)

shape = plt.Circle((0,0), radius = np.pi, color = 'purple', alpha= 0.4)

plt.grid(True)

plt.axis('scaled')

plt.xlim(-6,6)

plt.ylim(-6,6)

plt.title('circle')

plt.gca().add_patch(shape)

plt.subplot(122)

A = np.random.standard_normal((50, 2))

A += np.array((-1, -1))

B = np.random.standard_normal((50, 2))

B += np.array((1, 1))

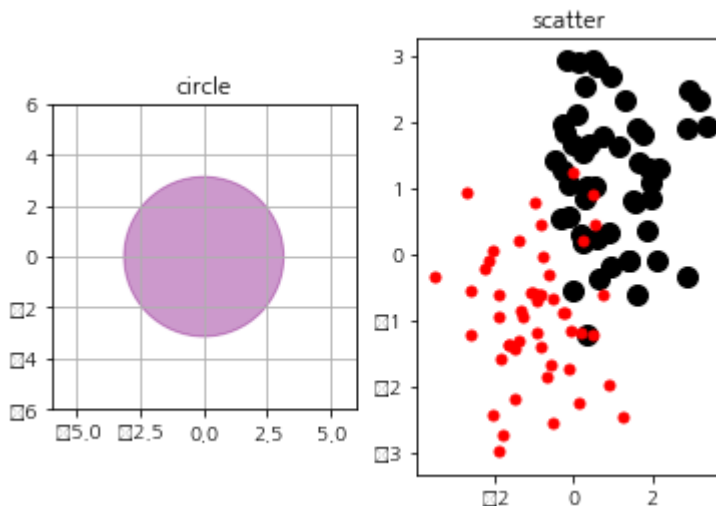
plt.scatter(B[:,0], B[:,1], c = 'k', s = 100.)

plt.scatter(A[:,0], A[:,1], c = 'r', s = 25.)

plt.title('scatter')

Out[47]:

Text(0.5, 1.0, 'scatter')



소감 2줄

연습문제 1과 2는 학습하면서 배운내용만으로 접근이 용이했기에 할만하였지만 연습문제 3은 이제 matplotlib에서 가져온 예제는 실로 당황스러웠습니다.

모르는 api만있는 것은 아니지만 parameter값을 매우 다르게 쓰여서 읽는데 애로사항이 많았습니다. 좀 익숙해질 필요를 많이 느낍니다.

