matplotlib tutorial 정리

matplotlib.pyplot

MATLAB 처럼 matplotlib 을 사용할 수 있게 하는 명령 스타일 함수들(command style functions) 모음이다.

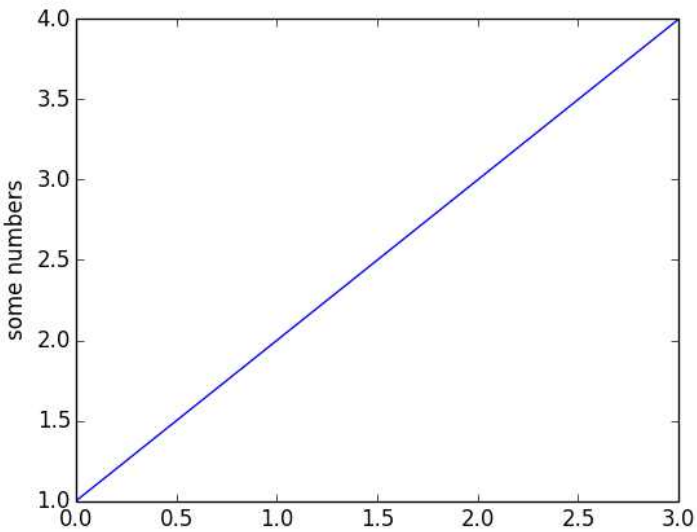
ex) figure 생성하고 -> figure에 plotting area 생성 -> plotting area 에 line 그리고 -> label 등을 꾸미고.. etc.

* plotting 함수들은 current axes 에 작용한다.

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```

** plot() 에 값 한개만 입력시에는 y 값으로 생각한다.

※ x 값은 자동생성한다.(0 부터 시작함)




```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16], 'ro')
plt.axis([0, 6, 0, 20])    # [xmin, xmax, ymin, ymax]
plt.show()
```

```
matplotlib.pyplot.plot(*args, **kwargs)
Plot lines and/or markers to the Axes.
plot (x ,y )          # plot x and y using default line style and color
plot (x ,y ,'bo')  # plot x and y using blue circle markers
plot (y )              # plot y using x as index array 0..N-1
plot (y ,'r+')       # ditto, but with red plusses
a .plot (x1 ,y1 ,'g^',x2 ,y2 ,'g-')
```

format string characters are accepted to control the line style or marker:

character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker
‘b’	blue
‘g’	green
‘r’	red
‘c’	cyan
‘m’	magenta
‘y’	yellow
‘k’	black
‘w’	white

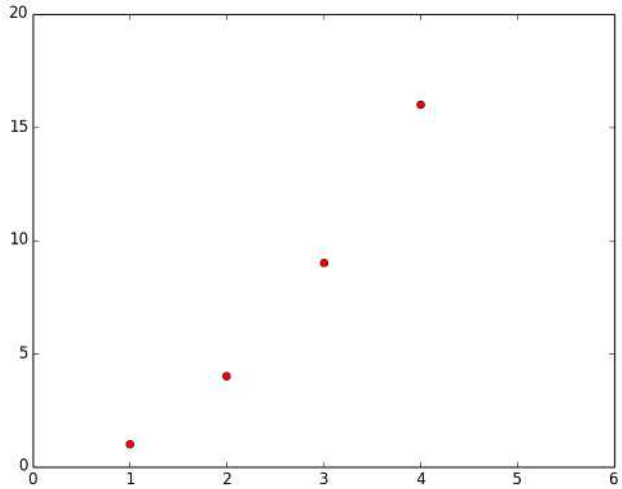
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.axis

matplotlib.pyplot.axis(*v, **kwargs)

Convenience method to get or set axis properties.

>>> axis (v)

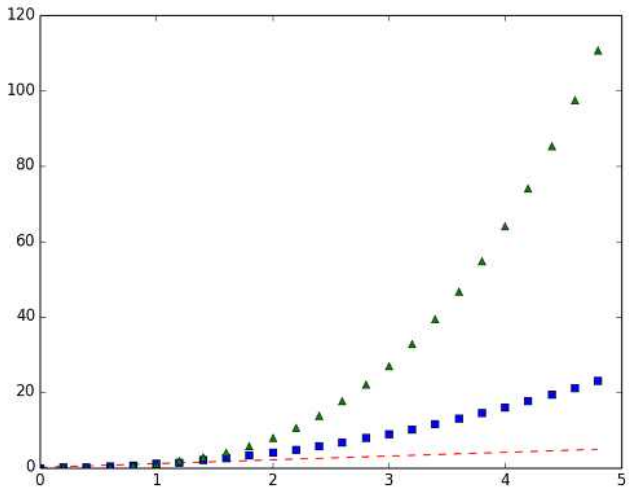
sets the min and max of the x and y axes, with v = [xmin, xmax, ymin, ymax].:



*** matplotlib 은 numpy array 를 사용한다.

※ seqence 자료 (list etc)는 내부적으로 numpy arra y로 변환 하여 사용한다.

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

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



line 속성 제어하기 (Controlling line properties)

참고 : matplotlib.lines.Line2D

http://matplotlib.org/1.3.1/api/artist_api.html#matplotlib.lines.Line2D

```
plt .plot (x ,y ,linewidth =2.0 )

line ,=plt .plot (x ,y ,'-')
line .set_antialiased (False )# turn off antialising
```

* setp() 명령어 사용

```
lines =plt .plot (x1 ,y1 ,x2 ,y2 )
# use keyword args

plt .setp (lines ,color ='r',linewidth =2.0 )
# or MATLAB style string value pairs

plt .setp (lines ,'color','r','linewidth',2.0 )
```

* line 속성 얻을때도, setp() 명령어 사용

```
lines =plt .plot ([1 ,2 ,3 ])
plt .setp (lines )
alpha: float
animated: [True | False]
antialiased or aa: [True | False]
. ..snip
```

여러개의 figure &axes 로 작업하기 (Working with multiple figures and axes)

pyplot --- current figure, current axes 의 개념이 있다.

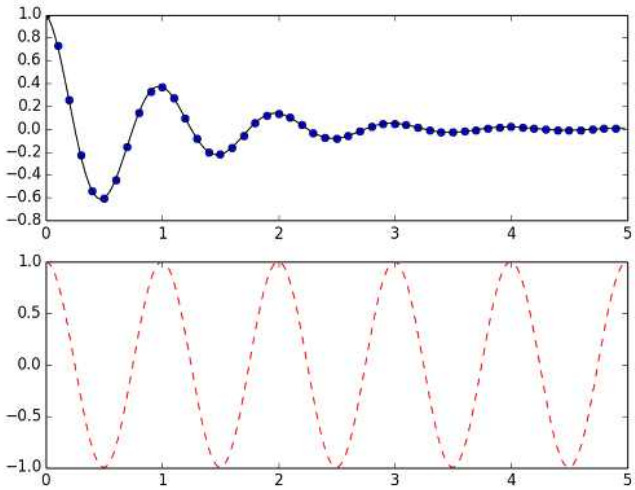
* 모든 plotting 명령어는 current axes 에 적용된다.
  * gca() --- current axes 반환 (matplotlib.axes.Axes instance)
  * gcf() --- current figure 반환 (matplotlib.figure.Figure instance)

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

def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.figure(1)   #  생략 가능
plt.subplot(211)   # row number, column number, figure number
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')

plt.subplot(212)
```

```
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
plt.show()
```

※ plt.figure(1) 생략 가능 : defalut 로 figure(1) 생성 된다.

※ axes 지정 하지않으면, defalut 로 subplot(111) 생성됨.



*** figure() 를 여러번 호출하여, 여러개의 figure 생성 가능하다.

(각 figure 는 자신만의 axes , subplot 를 갖는다)

```
import matplotlib.pyplot as plt
plt.figure(1)                # the first figure
plt.subplot(211)             # the first subplot in the first figure
plt.plot([1,2,3])
plt.subplot(212)             # the second subplot in the first figure
plt.plot([4,5,6])


plt.figure(2)                # a second figure
plt.plot([4,5,6])            # creates a subplot(111) by default

plt.figure(1)                # figure 1 current; subplot(212) still current
plt.subplot(211)             # make subplot(211) in figure1 current
plt.title('Easy as 1,2,3')   # subplot 211 title
```

* clf() --- clear current figure.

* cla() --- clear current axes.

*** 메모리 해제 : close()

※ close() 호출되기 전까지는, pyplot 이 참조를 계속 가지고 있다.

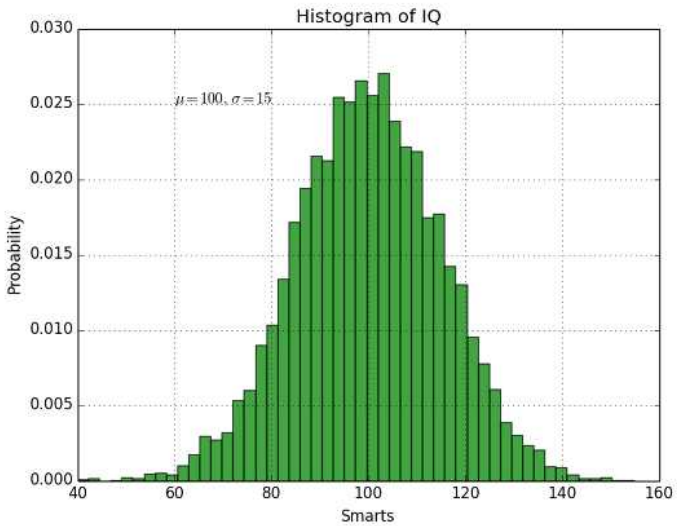
텍스트 작업하기 (Working with text)

http://matplotlib.org/1.3.1/api/pyplot_api.html#matplotlib.pyplot.text

matplotlib.pyplot.text(x, y , s , fontdict=None , withdash=False , **kwargs)

-- 좌표 x,y 에 텍스트 s 출력하기

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

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 50, normed=1, facecolor='g', alpha=0.75)


plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```




```
t =plt .xlabel ('my data',fontsize =14 ,color ='red')
```

텍스트로 수학식 표현하기 (Using mathematical expressions in text)

built-in TeX expression parser and layout engine

$\sigma_i = 15$

```
plt .title (r'$\sigma_i=15$')
```

참고 : http://matplotlib.org/1.3.1/users/mathtext.html#mathtext-tutorial

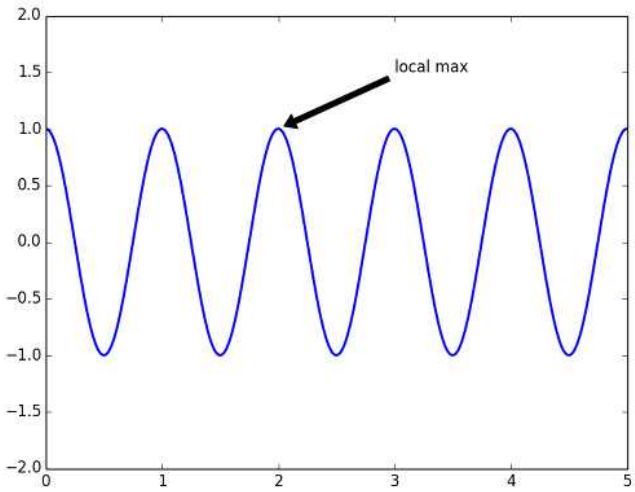
텍스트 주석달기 (Annotating text)

참고 : http://matplotlib.org/1.3.1/api/pyplot_api.html#matplotlib.pyplot.annotate

matplotlib.pyplot.annotate(*args, **kwargs)

```
annotate (s ,xy ,xytext =None ,xycoords ='data',
          textcoords ='data',arrowprops =None ,**kwargs )
---  주석 = s, 화살표 시작 = x,y ,  주석 텍스트 시작 = xytext
```

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

ax = plt.subplot(111)

t = np.arange(0.0, 5.0, 0.01)
s = np.cos(2*np.pi*t)
line, = plt.plot(t, s, lw=2)

plt.annotate('local max', xy=(2, 1), xytext=(3, 1.5),
             arrowprops=dict(facecolor='black', shrink=0.05),
             )
plt.ylim(-2,2)
plt.show()
```



출처: http://freeprog.tistory.com/15 [취미로 하는 프로그래밍 !!!]