MATPLOTLIB 기초 이해하기

Moon Yong Joon

MATPLOTLIB PYPLOT 기초

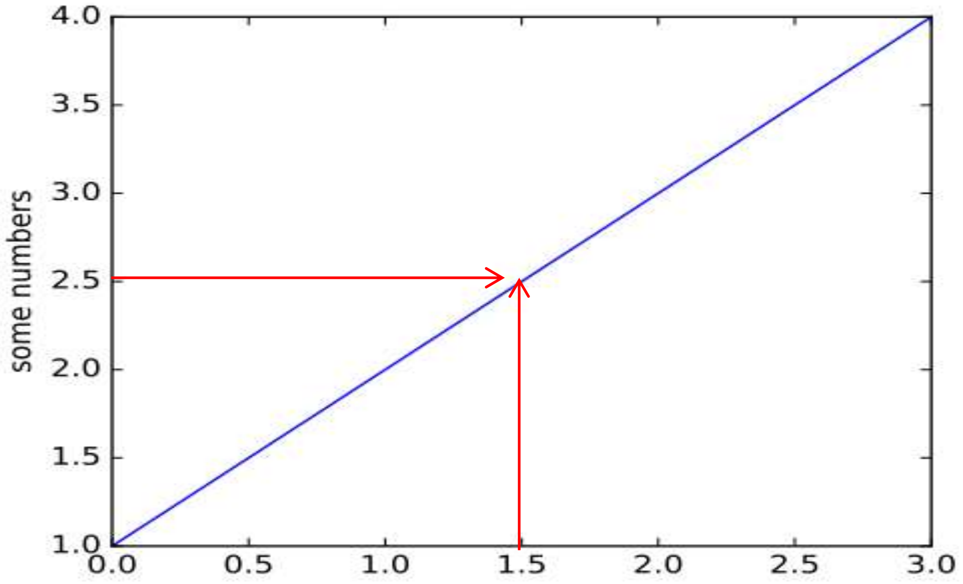
Moon Yong Joon

좌표이해하기

좌표

그래프는 일단 x,y축 좌표에 대해 이해를 해야 함

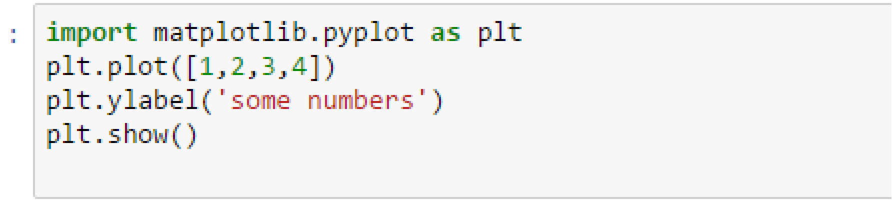
Y축



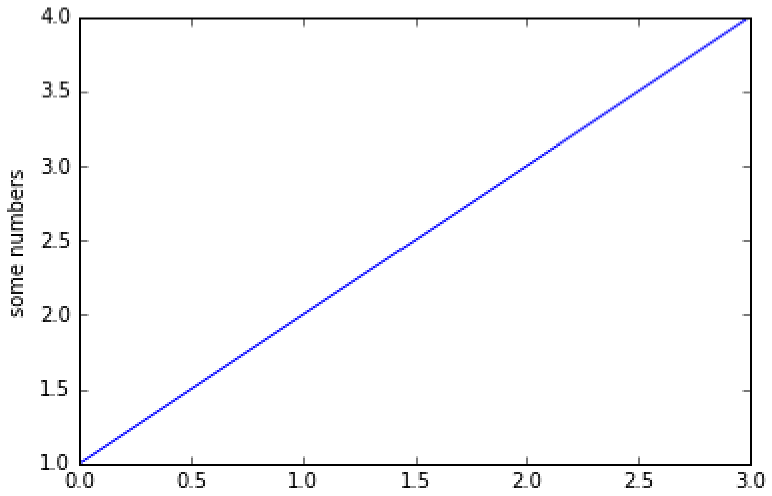
X축

좌표 기준

matplotlib은 하나의 리스크만 넣으면 리스트의 index가 x축, 값이 y축으로 인식해서 그래프 표시



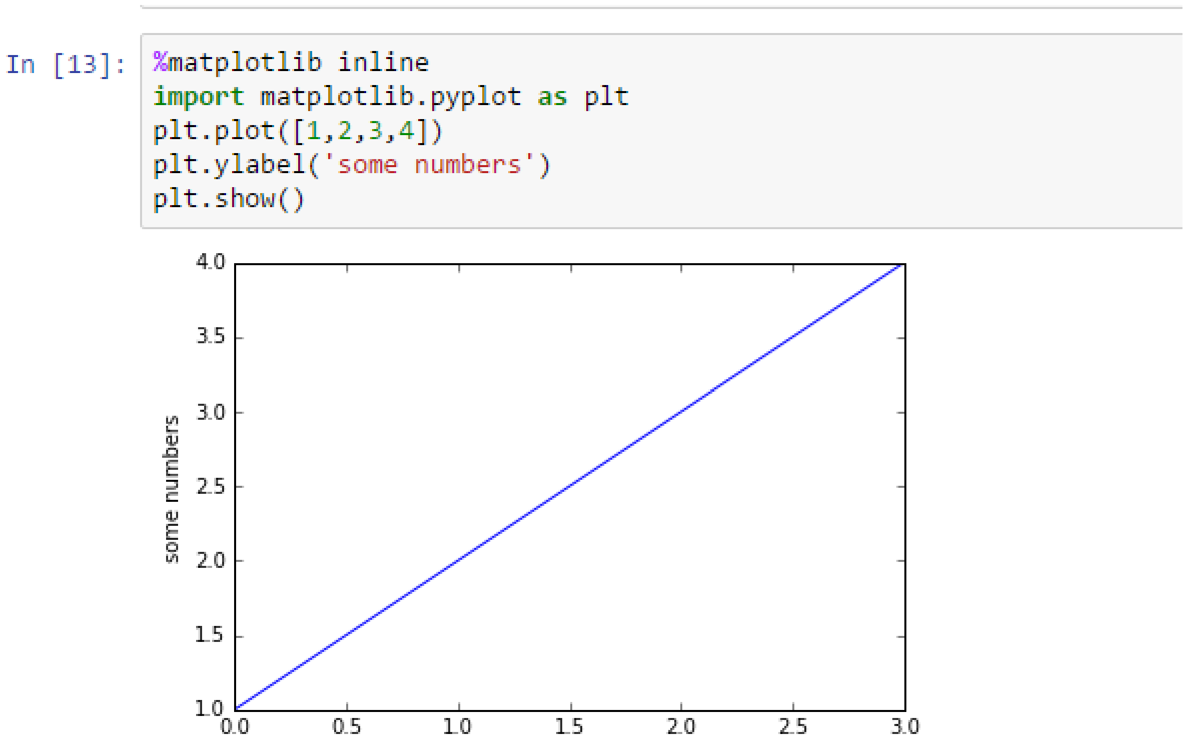
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```



jupyter 내에서 그래프 보기

jupyter notebook 실행

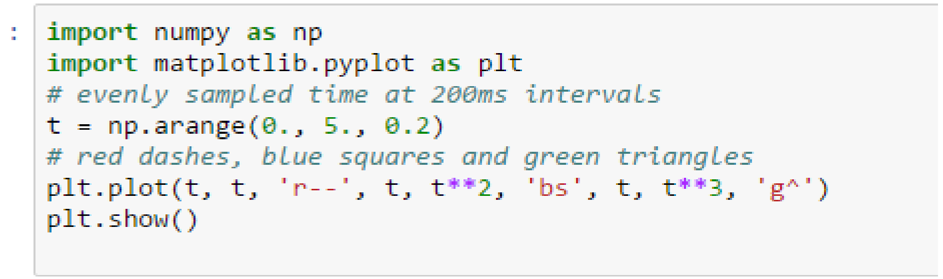
%matplotlib inline 명령을 먼저 실행해야 jupyter notebook 내에서 그래프가 보임



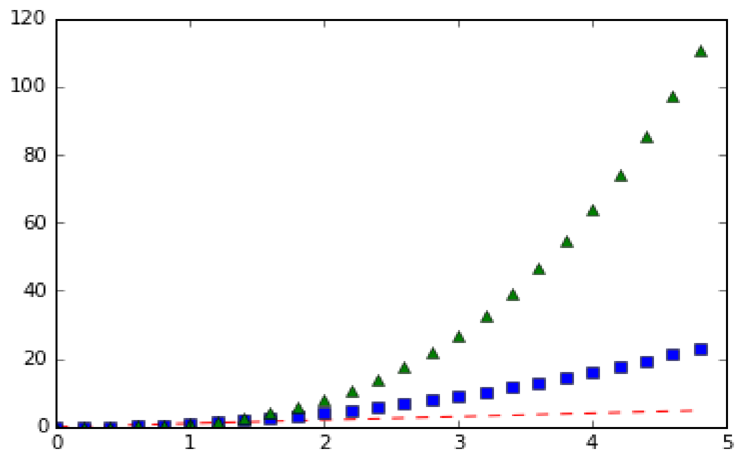
```
In [13]: %matplotlib inline
         import matplotlib.pyplot as plt
         plt.plot([1,2,3,4])
         plt.ylabel('some numbers')
         plt.show()
```

여러 개의 선을 그리기

plot 함수에 x,y,그래프색과모양 3개를 묶어서 3개를 표현해서 그리기



```
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()
```

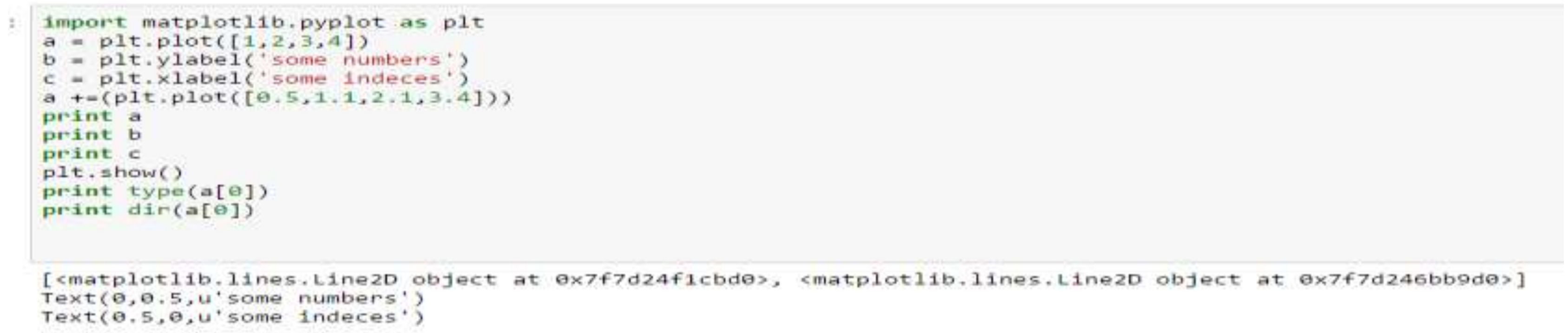


y = x
y = x**2
y = x**3
에 대한 함수의 그래프를 표현

Plot 구조 이해하기

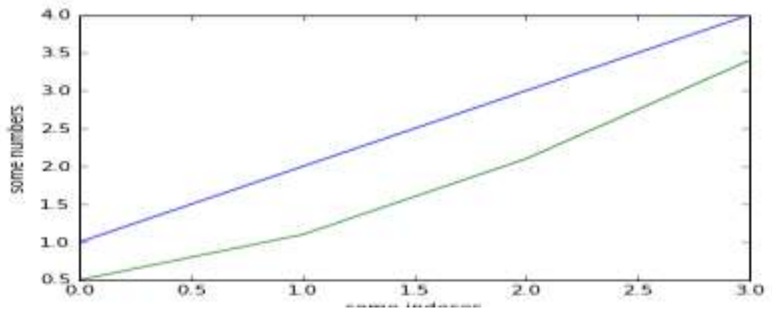
내부 구조 이해하기1 : Line2D

plot 함수를 실행하면 하나의 list가 생기고 Line2D object가 생기며 plot함수를 하나더 실행해서 기존 list에 원소로 추가 가능



```
import matplotlib.pyplot as plt
a = plt.plot([1,2,3,4])
b = plt.ylabel('some numbers')
c = plt.xlabel('some indeces')
a +=(plt.plot([0.5,1.1,2.1,3.4]))
print a
print b
print c
plt.show()
print type(a[0])
print dir(a[0])
```

```
[<matplotlib.lines.Line2D object at 0x7f7d24f1cbd0>, <matplotlib.lines.Line2D object at 0x7f7d246bb9d0>]
Text(0,0.5,u'some numbers')
Text(0.5,0,u'some indeces')
```

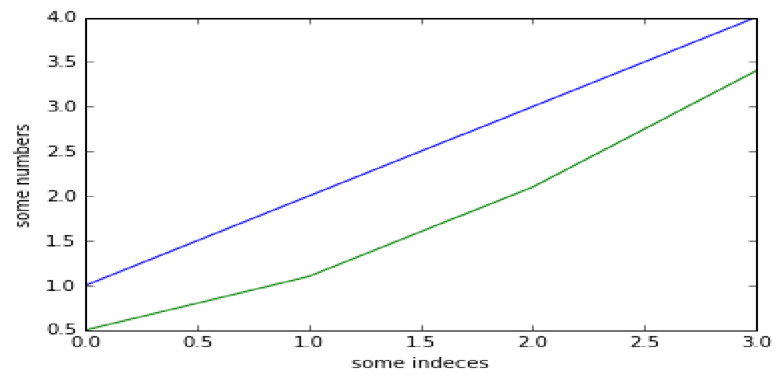


내부 구조 이해하기 2 : Line2D

2개의 plot 함수를 실행해서 출력해도 앞장의 경우와 동일하게 출력됨

```
import matplotlib.pyplot as plt
a = plt.plot([1,2,3,4])
b = plt.ylabel('some numbers')
c = plt.xlabel('some indeces')
d = plt.plot([0.5,1.1,2.1,3.4])
print a
print b
print c
print d
plt.show()
```

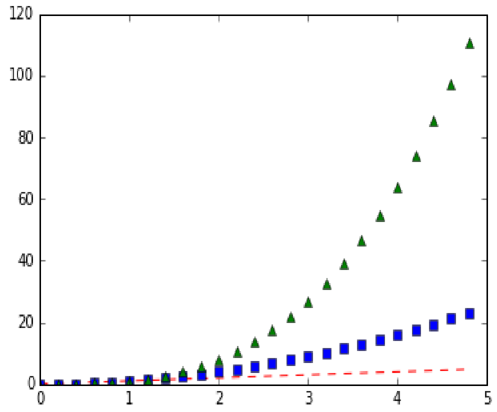
```
[<matplotlib.lines.Line2D object at 0x7f7d2410af50>]
Text(0,0.5,u'some numbers')
Text(0.5,0,u'some indeces')
[<matplotlib.lines.Line2D object at 0x7f7d24685a10>]
```



내부 구조 이해하기 3 : Line2D

plot 함수에 3개의 그래프를 연속해서 정의하고 실행하면 list에 Line2D가 3개 생겨서 표시

```
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
a = plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
print a
```

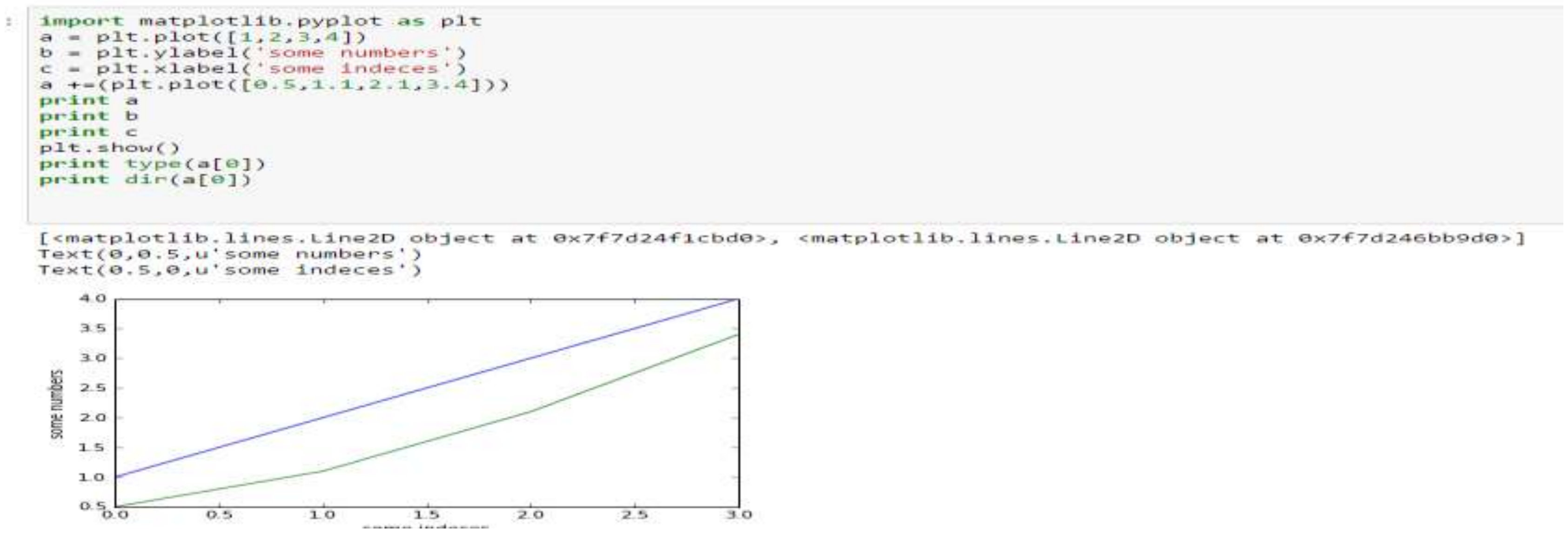


```
[<matplotlib.lines.Line2D object at 0x7f7d24203f50>, <matplotlib.lines.Line2D object at 0x7f7d241de150>, <matplotlib.lines.Line
2D object at 0x7f7d241de7d0>]
```

Text 구조 이해하기

내부 구조 이해하기 :Text

ylabel함수를 실행하면 하나의 Text object 가 생김



```
import matplotlib.pyplot as plt
a = plt.plot([1,2,3,4])
b = plt.ylabel('some numbers')
c = plt.xlabel('some indeces')
a +=(plt.plot([0.5,1.1,2.1,3.4]))
print a
print b
print c
plt.show()
print type(a[0])
print dir(a[0])
```

```
[<matplotlib.lines.Line2D object at 0x7f7d24f1cbd0>, <matplotlib.lines.Line2D object at 0x7f7d246bb9d0>]
Text(0,0.5,u'some numbers')
Text(0.5,0,u'some indeces')
```

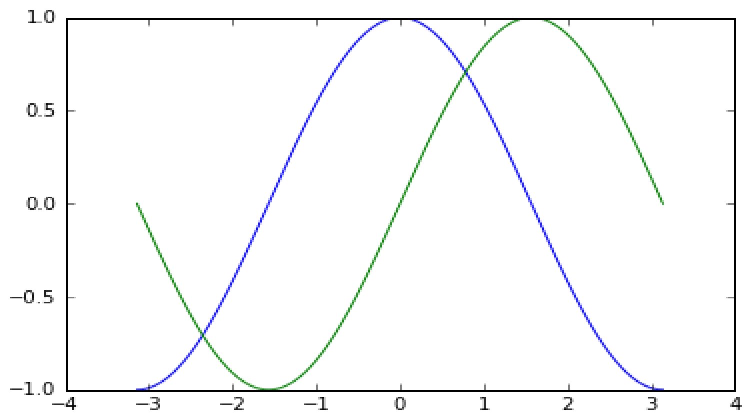
그래프/캔버스 이해하기

그래프 이해하기

하나의 캔버스에 두개의 그래프 처리

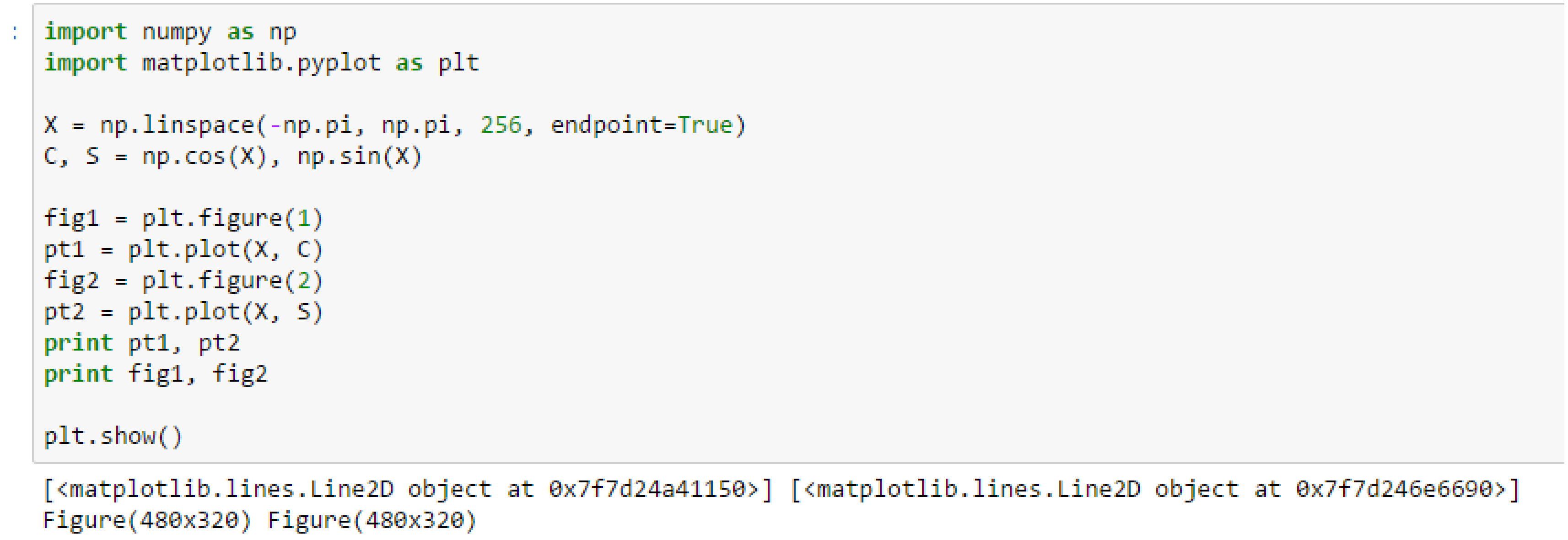
```python
import numpy as np
import matplotlib.pyplot as plt

X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)

pt1 = plt.plot(X, C)
pt2 = plt.plot(X, S)
print pt1, pt2

plt.show()
```

```
[<matplotlib.lines.Line2D object at 0x7f7d24028b10>] [<matplotlib.lines.Line2D object at 0x7f7d240d2190>]
```



캔버스 이해하기 1

Figure 객체를 두개 만들고 내부에 처리

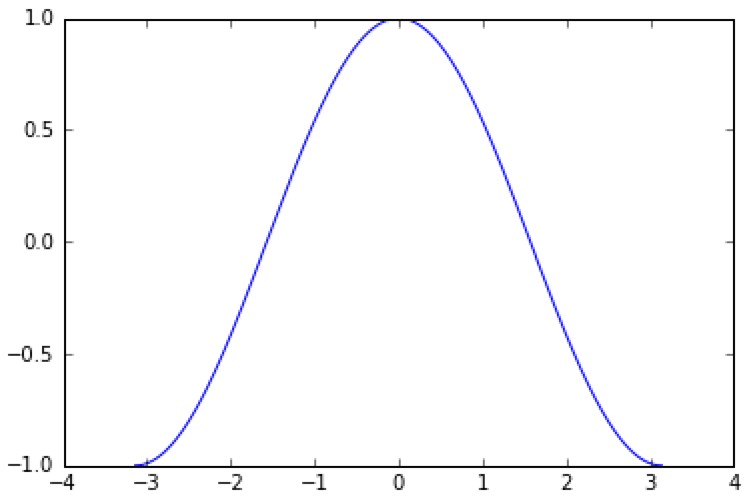


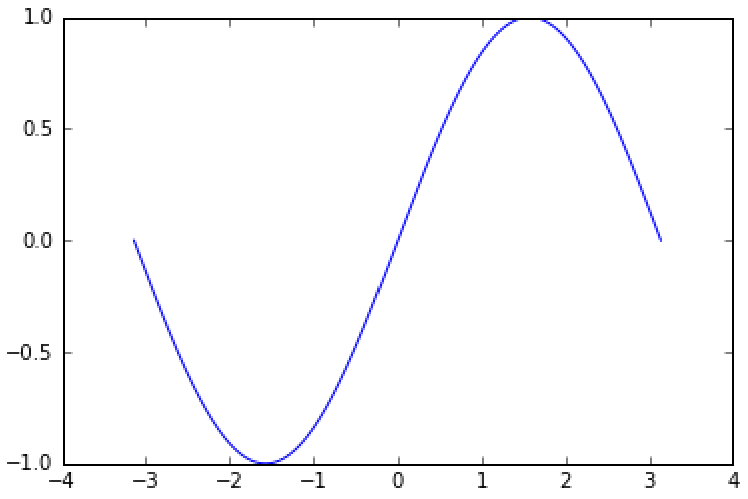
```
:  import numpy as np
   import matplotlib.pyplot as plt

   X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
   C, S = np.cos(X), np.sin(X)

   fig1 = plt.figure(1)
   pt1 = plt.plot(X, C)
   fig2 = plt.figure(2)
   pt2 = plt.plot(X, S)
   print pt1, pt2
   print fig1, fig2

   plt.show()
```

```
[<matplotlib.lines.Line2D object at 0x7f7d24a41150>] [<matplotlib.lines.Line2D object at 0x7f7d246e6690>]
Figure(480x320) Figure(480x320)
```

캔버스 이해하기 2

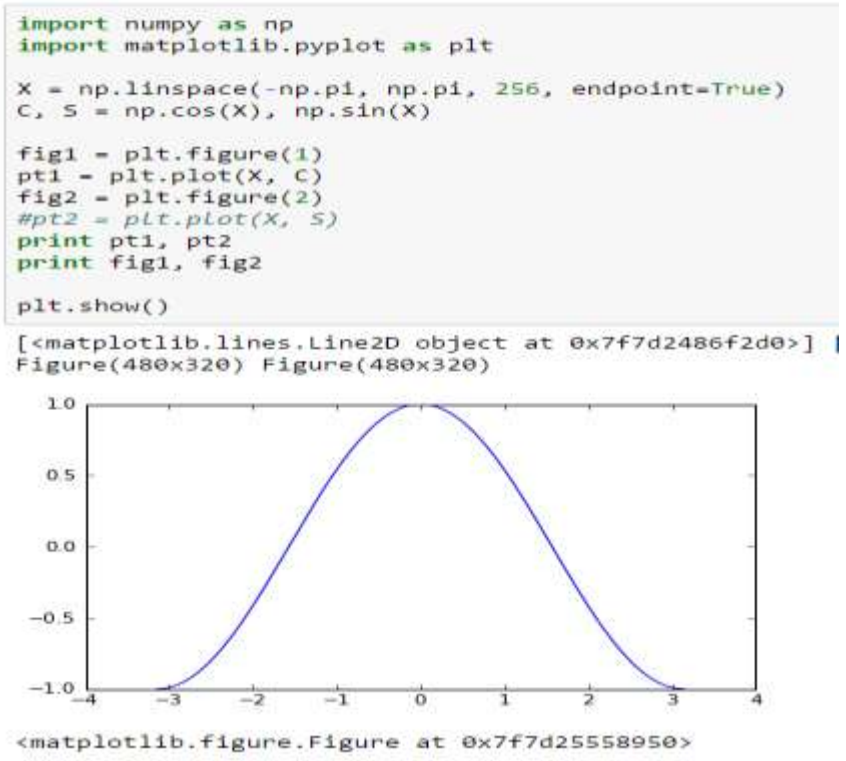
Figure 클래스별로 별도 캔버스를 구성





캔버스 이해하기 3

하나의 plot를 주석처리하면 캔버스 하나는 출력되지만 다른 하나는 객체 주소만 출력



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

X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)

fig1 = plt.figure(1)
pt1 = plt.plot(X, C)
fig2 = plt.figure(2)
#pt2 = plt.plot(X, S)
print pt1, pt2
print fig1, fig2

plt.show()
```

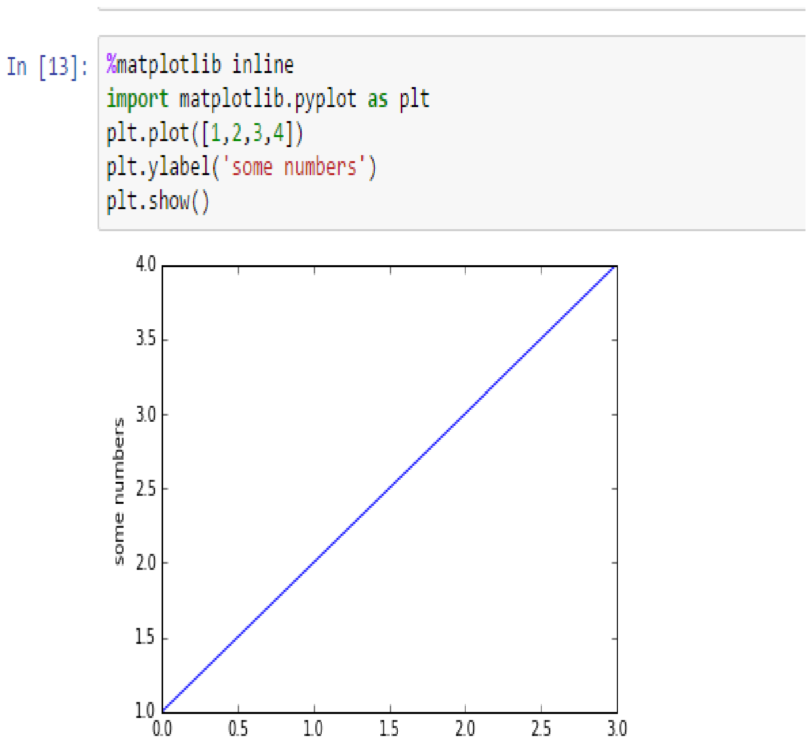
```
[<matplotlib.lines.Line2D object at 0x7f7d2486f2d0>]
Figure(480x320) Figure(480x320)
```

```
<matplotlib.figure.Figure at 0x7f7d25558950>
```

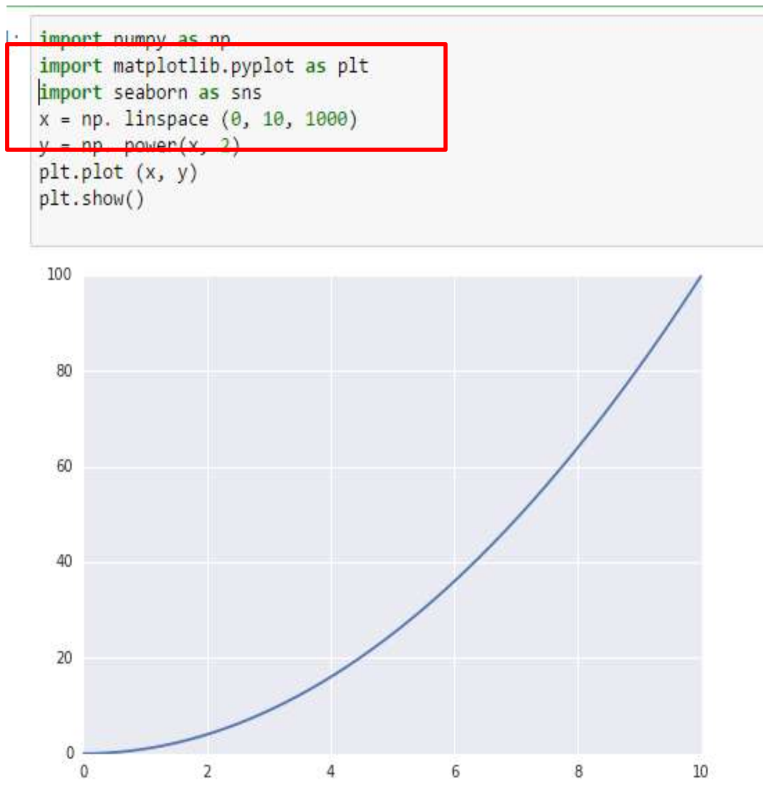
Seaborn 꾸미기

Seaborn 적용 꾸미기

seaborn을 이용하면 그래프의 격자가 꾸며짐



```
In [13]: %matplotlib inline
         import matplotlib.pyplot as plt
         plt.plot([1,2,3,4])
         plt.ylabel('some numbers')
         plt.show()
```

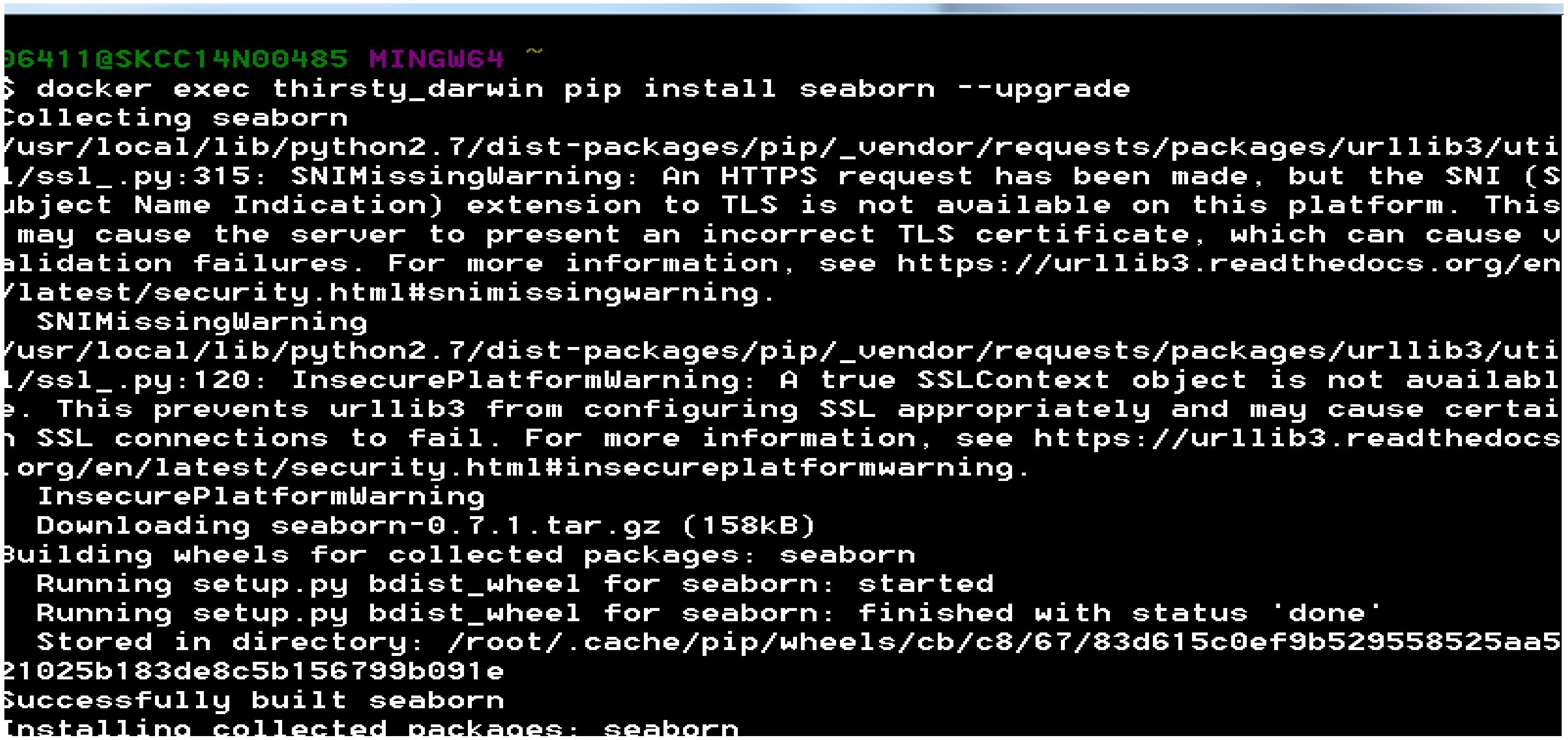


```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
x = np. linspace (0, 10, 1000)
y = np. power(x, 2)
plt.plot (x, y)
plt.show()
```

Seaborn install

docker에서 seaborn을 pip로 설치

docker exec {도커이미지} pip install seaborn --upgrade



```
96411@SKCC14N00485 MINGW64 ~
$ docker exec thirsty_darwin pip install seaborn --upgrade
Collecting seaborn
/usr/local/lib/python2.7/dist-packages/pip/_vendor/requests/packages/urllib3/uti
l/ssl_.py:315: SNIMissingWarning: An HTTPS request has been made, but the SNI (S
ubject Name Indication) extension to TLS is not available on this platform. This
 may cause the server to present an incorrect TLS certificate, which can cause v
alidation failures. For more information, see https://urllib3.readthedocs.org/en
/latest/security.html#snimissingwarning.
  SNIMissingWarning
/usr/local/lib/python2.7/dist-packages/pip/_vendor/requests/packages/urllib3/uti
l/ssl_.py:120: InsecurePlatformWarning: A true SSLContext object is not availabl
e. This prevents urllib3 from configuring SSL appropriately and may cause certai
n SSL connections to fail. For more information, see https://urllib3.readthedocs
.org/en/latest/security.html#insecureplatformwarning.
  InsecurePlatformWarning
  Downloading seaborn-0.7.1.tar.gz (158kB)
Building wheels for collected packages: seaborn
  Running setup.py bdist_wheel for seaborn: started
  Running setup.py bdist_wheel for seaborn: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/cb/c8/67/83d615c0ef9b529558525aa5
21025b183de8c5b156799b091e
Successfully built seaborn
Installing collected packages: seaborn
```

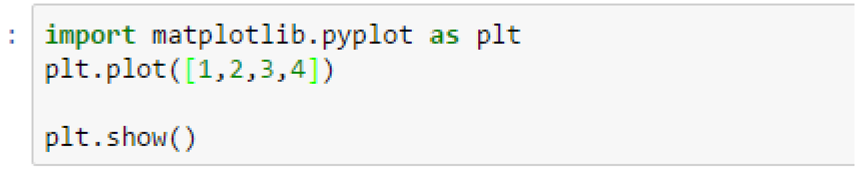
MATPLOTLIB PYPLOT PLOT 함수

Moon Yong Joon

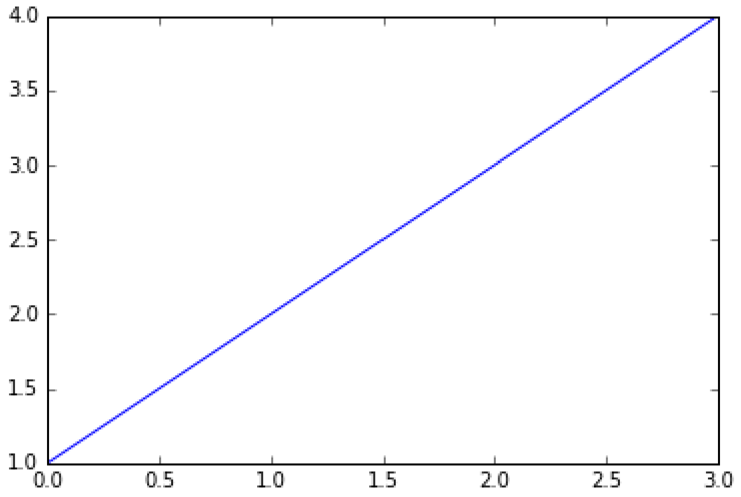
선 그래프

plot 함수 : 한축만 1

y축은 plot 함수 내의 값이고 x 축은 인덱스를 표시

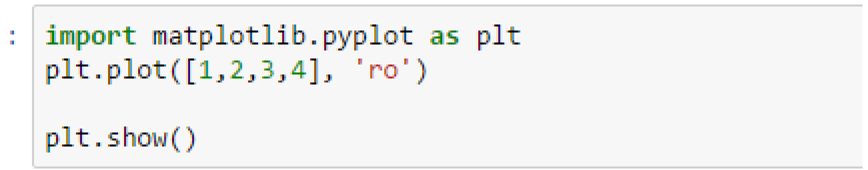


```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])

plt.show()
```

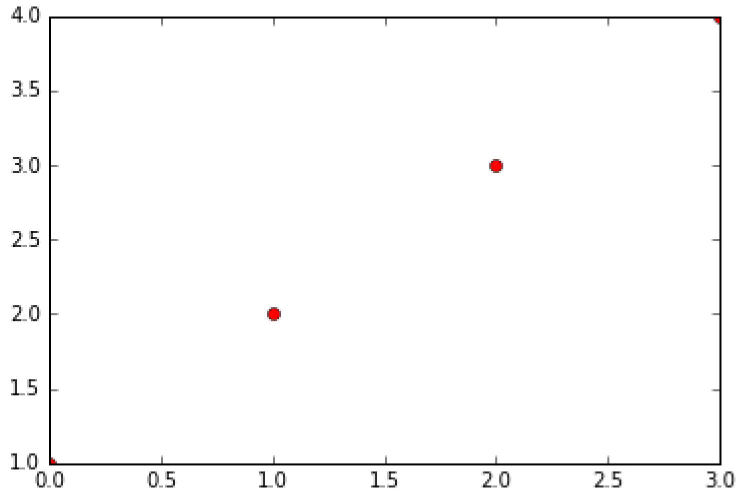


plot 함수 : 한축만 2

y축은 plot 함수 내의 값이고 x 축은 인덱스를 표시를 marker를 넣고 확인

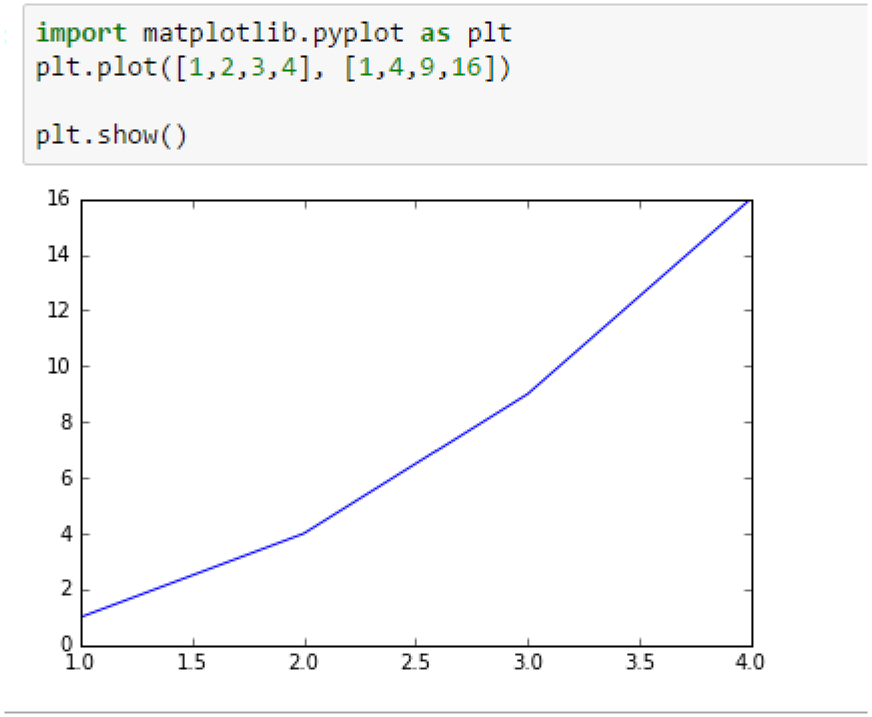


```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], 'ro')

plt.show()
```



plot 함수 : x축과 y축 1

x축과 y축 넣고 그래프 보기



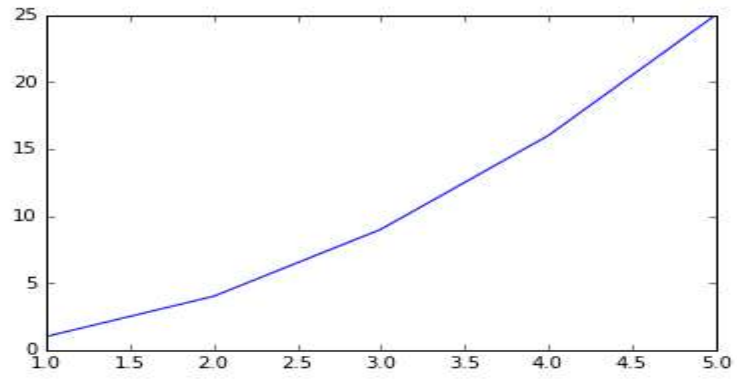
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16])

plt.show()
```

plot 함수 : x축과 y축 2

x축과 y축 넣고 그래프 보기

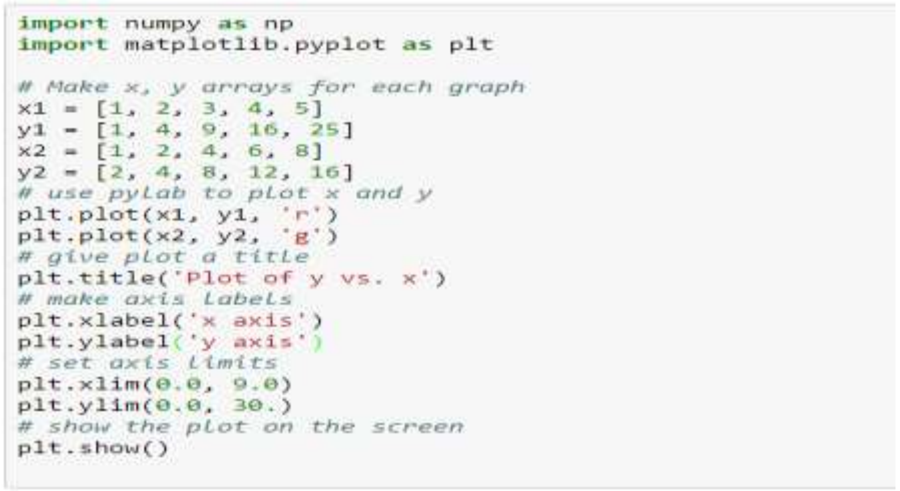
```python
import numpy as np
import matplotlib.pyplot as plt

# Make an array of x values
x = [1, 2, 3, 4, 5]
# Make an array of y values for each x value
y = [1, 4, 9, 16, 25]
# use pylab to plot x and y
mo = plt.plot(x, y)
print mo
# show the plot on the screen
plt.show()
```

```
[<matplotlib.lines.Line2D object at 0x7f7d24a2d2d0>]
```

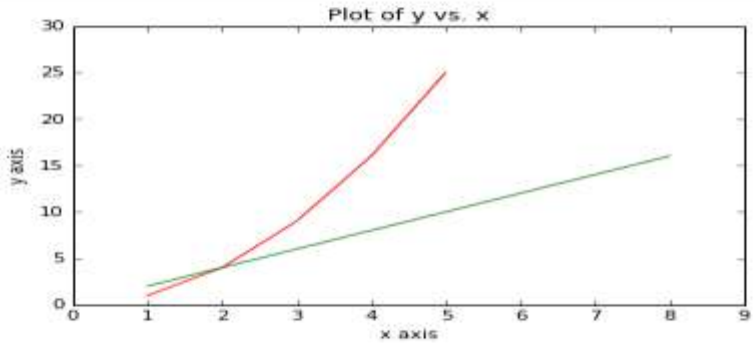


plot 함수 :두개 사용

plot 함수를 두번 사용해서 2개의 그래프 보기



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

# Make x, y arrays for each graph
x1 = [1, 2, 3, 4, 5]
y1 = [1, 4, 9, 16, 25]
x2 = [1, 2, 4, 6, 8]
y2 = [2, 4, 8, 12, 16]
# use pylab to plot x and y
plt.plot(x1, y1, 'r')
plt.plot(x2, y2, 'g')
# give plot a title
plt.title('Plot of y vs. x')
# make axis labels
plt.xlabel('x axis')
plt.ylabel('y axis')
# set axis limits
plt.xlim(0.0, 9.0)
plt.ylim(0.0, 30.)
# show the plot on the screen
plt.show()
```



plot 함수 : marker 만들기

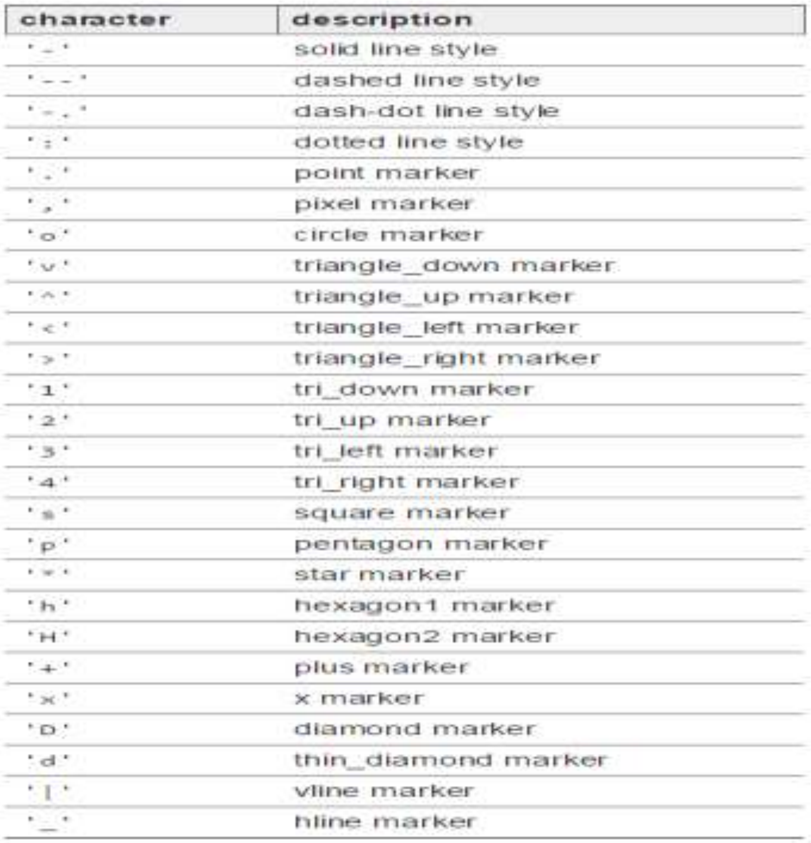
색상과 모양을 넣어 marker를 표시

색상

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

+

모양

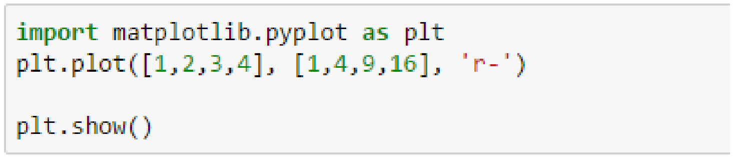


character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'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

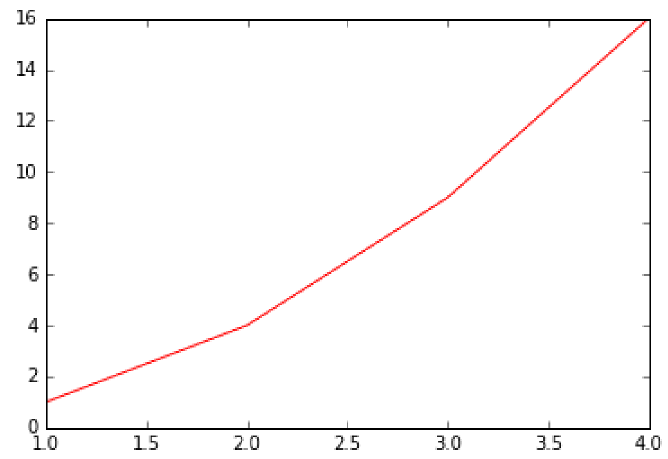
plot 함수 :marker 넣기 1

Plot 함수에 solid('r-'), dash('r—')를 넣고 그래프 그리기

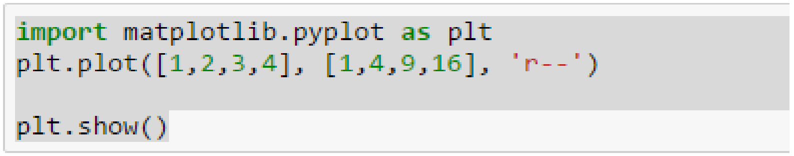
solid



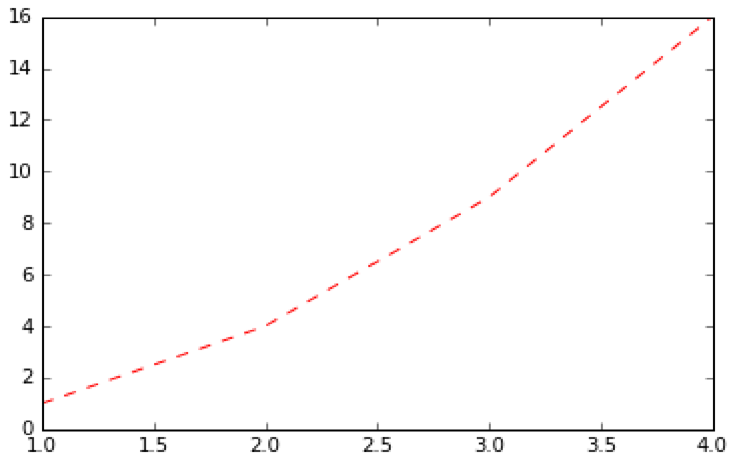
```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16], 'r-')

plt.show()
```



dash

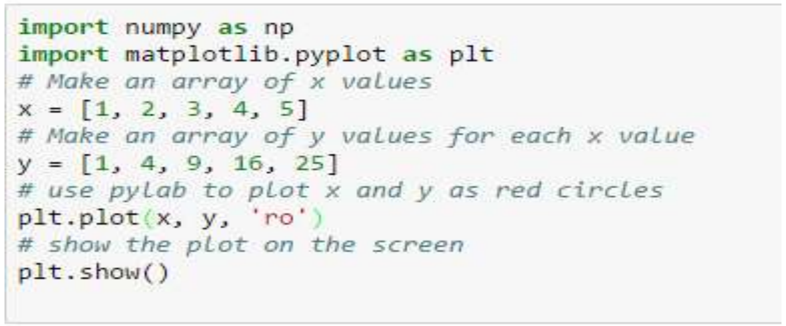


```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16], 'r--')

plt.show()
```

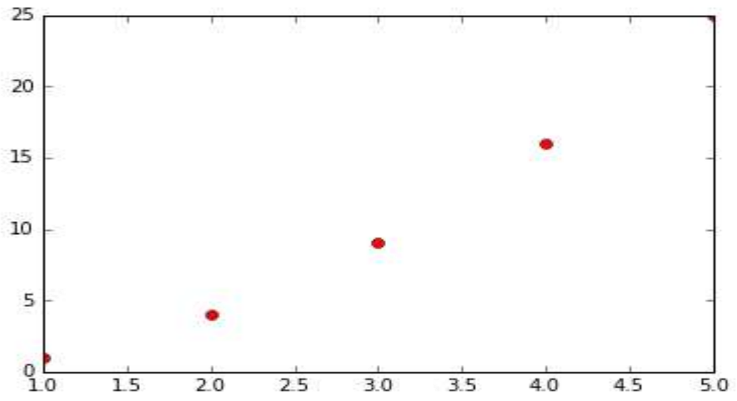


plot 함수: marker 넣기 2

plot 함수 파라미터에 circle marker(‘ro’)를 넣고 표시

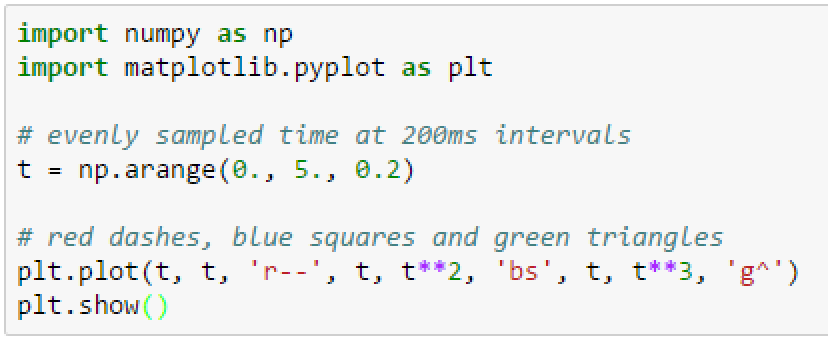


```
import numpy as np
import matplotlib.pyplot as plt
# Make an array of x values
x = [1, 2, 3, 4, 5]
# Make an array of y values for each x value
y = [1, 4, 9, 16, 25]
# use pylab to plot x and y as red circles
plt.plot(x, y, 'ro')
# show the plot on the screen
plt.show()
```

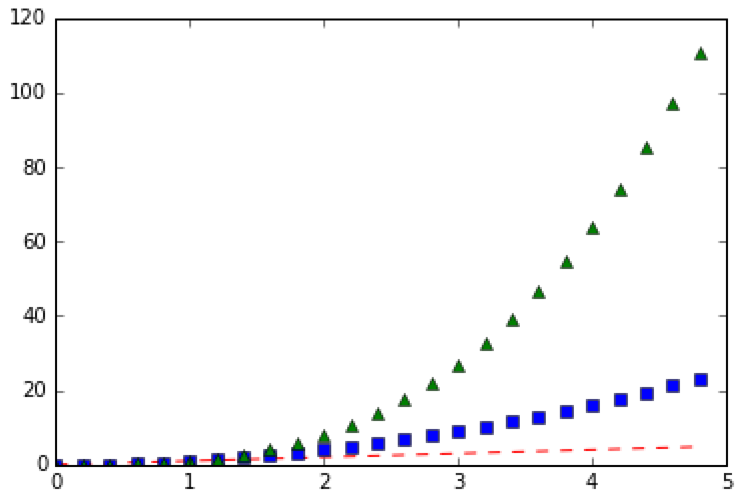


plot 함수 :marker 여러 개 넣기

Plot 함수에 marker 넣고 그래프 그리기



```
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()
```



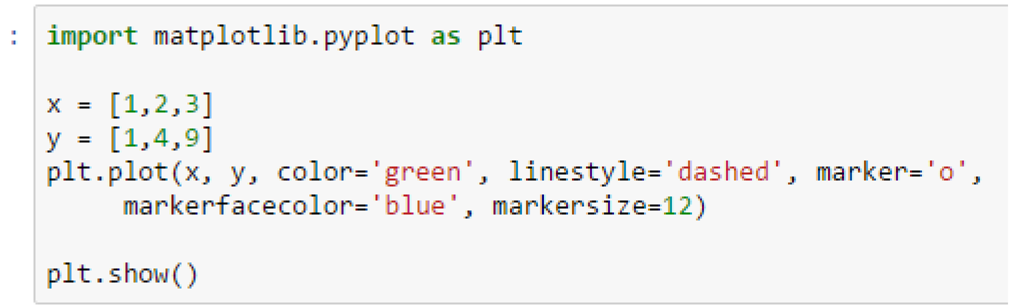
'g^' : green triangle

'bs' : blue square

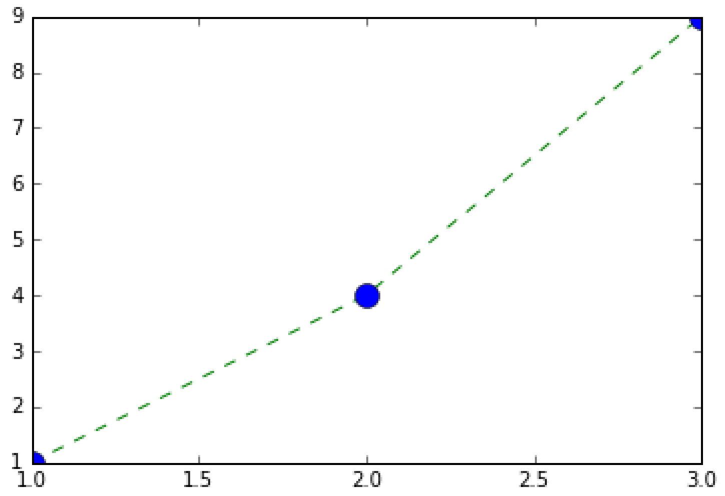
'r--' : red dash

plot 함수 : marker keyword

Plot 함수에 marker를 키워드 인자로 넣기

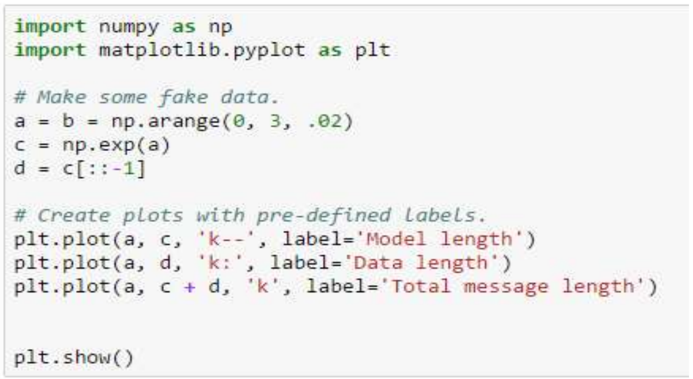


```
import matplotlib.pyplot as plt

x = [1,2,3]
y = [1,4,9]
plt.plot(x, y, color='green', linestyle='dashed', marker='o',
     markerfacecolor='blue', markersize=12)

plt.show()
```

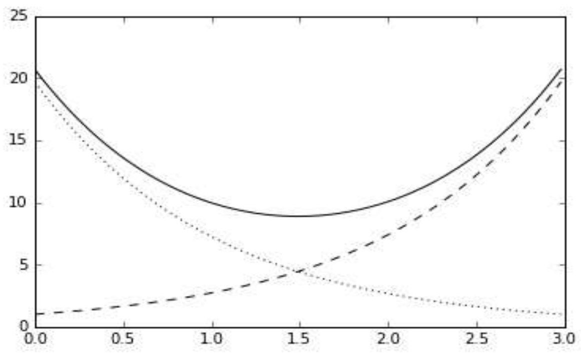


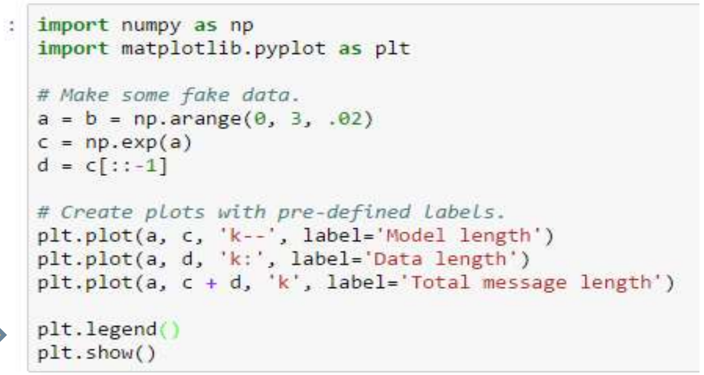
plot 함수 : label

Plot 함수에 legend함수 처리를 위해 label을 정의



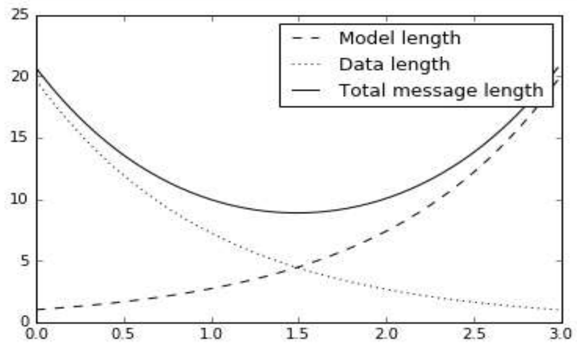
```
import numpy as np
import matplotlib.pyplot as plt

# Make some fake data.
a = b = np.arange(0, 3, .02)
c = np.exp(a)
d = c[::-1]

# Create plots with pre-defined labels.
plt.plot(a, c, 'k--', label='Model length')
plt.plot(a, d, 'k:', label='Data length')
plt.plot(a, c + d, 'k', label='Total message length')


plt.show()
```





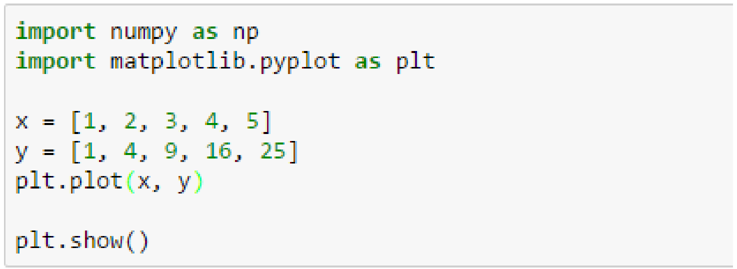
```
import numpy as np
import matplotlib.pyplot as plt

# Make some fake data.
a = b = np.arange(0, 3, .02)
c = np.exp(a)
d = c[::-1]

# Create plots with pre-defined labels.
plt.plot(a, c, 'k--', label='Model length')
plt.plot(a, d, 'k:', label='Data length')
plt.plot(a, c + d, 'k', label='Total message length')

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

legend 함수 호출하면 범주 표시

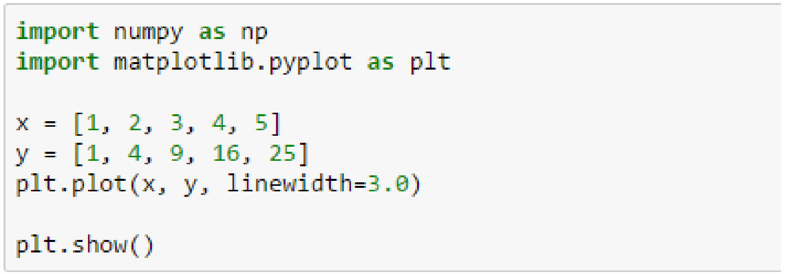


plot 함수 : linewidth

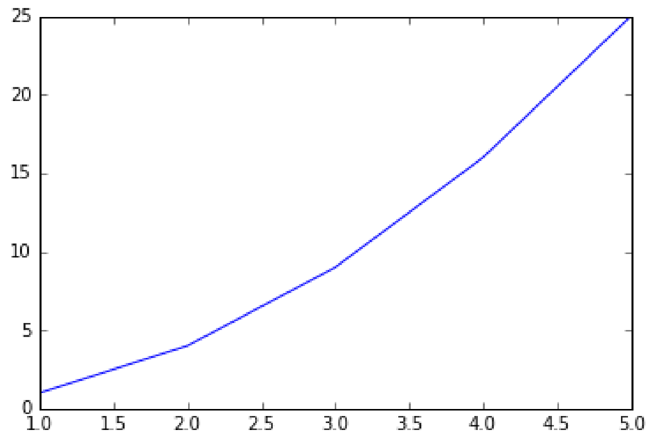
Plot 함수에 line을 굵게 하려면 linewidth에 값을 부여

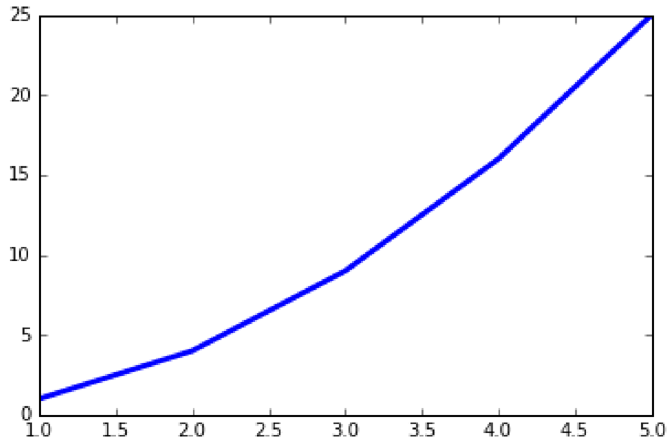


```
import numpy as np
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]
plt.plot(x, y)

plt.show()
```



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

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]
plt.plot(x, y, linewidth=3.0)

plt.show()
```

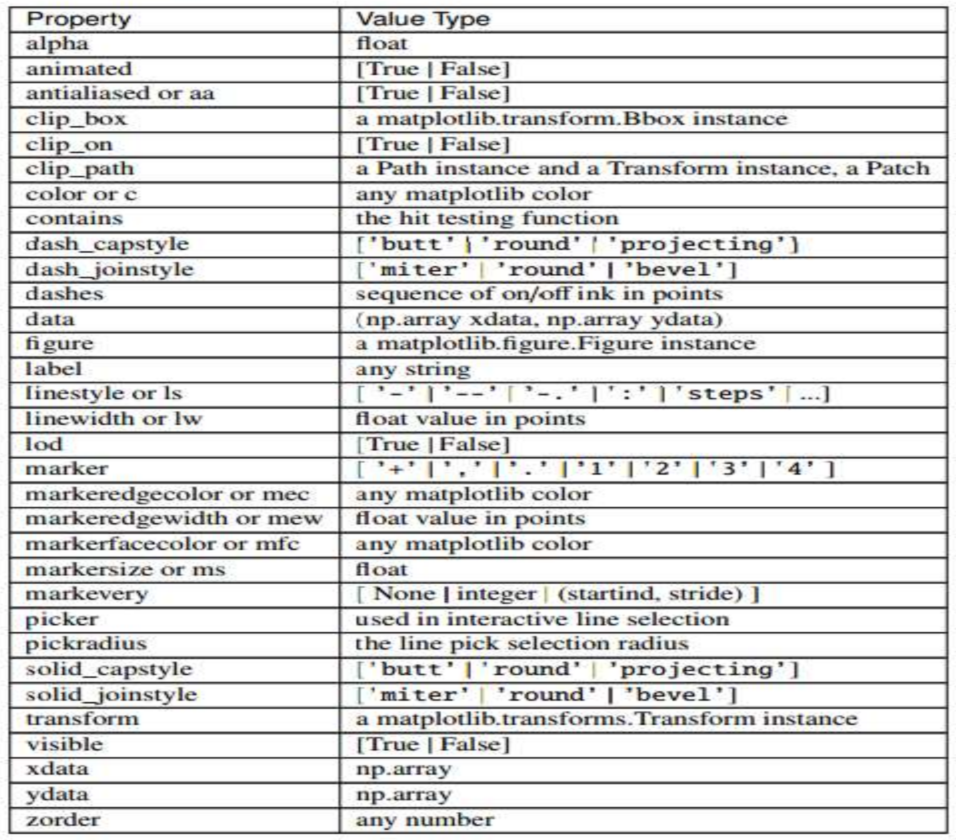




Line2D 클래스

Line2D property

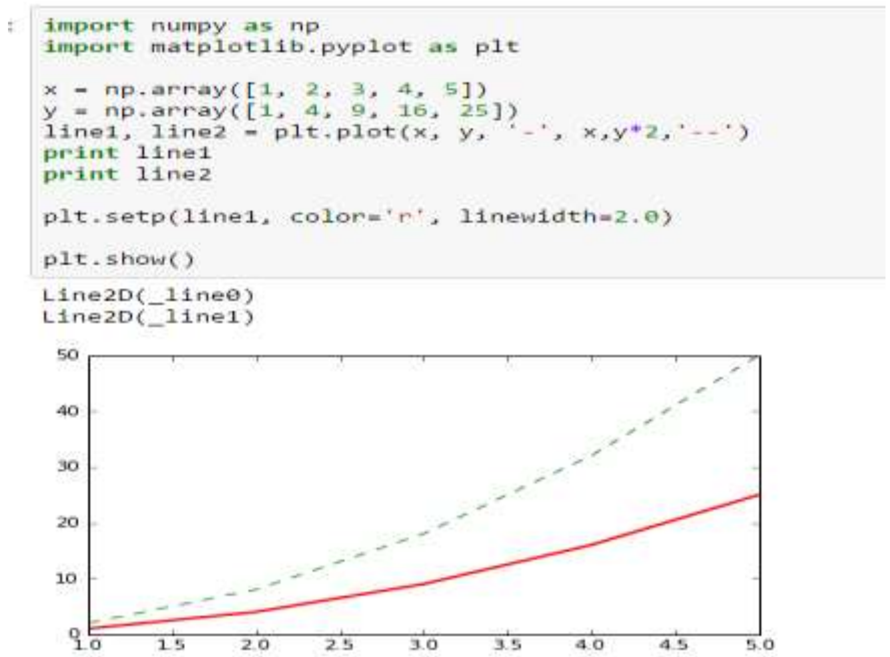
Line2D property



Property	Value Type
alpha	float
animated	[True | False]
antialiased or aa	[True | False]
clip_box	a matplotlib.transform.Bbox instance
clip_on	[True | False]
clip_path	a Path instance and a Transform instance, a Patch
color or c	any matplotlib color
contains	the hit testing function
dash_capstyle	['butt' | 'round' | 'projecting']
dash_joinstyle	['miter' | 'round' | 'bevel']
dashes	sequence of on/off ink in points
data	(np.array xdata, np.array ydata)
figure	a matplotlib.figure.Figure instance
label	any string
linestyle or ls	['-' | '--' | '-.' | ':' | 'steps' | ...]
linewidth or lw	float value in points
lod	[True | False]
marker	['+' | ',' | '.' | '1' | '2' | '3' | '4']
markeredgecolor or mec	any matplotlib color
markeredgewidth or mew	float value in points
markerfacecolor or mfc	any matplotlib color
markersize or ms	float
markevery	[None | integer | (startind, stride)]
picker	used in interactive line selection
pickradius	the line pick selection radius
solid_capstyle	['butt' | 'round' | 'projecting']
solid_joinstyle	['miter' | 'round' | 'bevel']
transform	a matplotlib.transforms.Transform instance
visible	[True | False]
xdata	np.array
ydata	np.array
zorder	any number

setp 함수로 Line2D 세팅하기

색상과 라인너비를 세팅



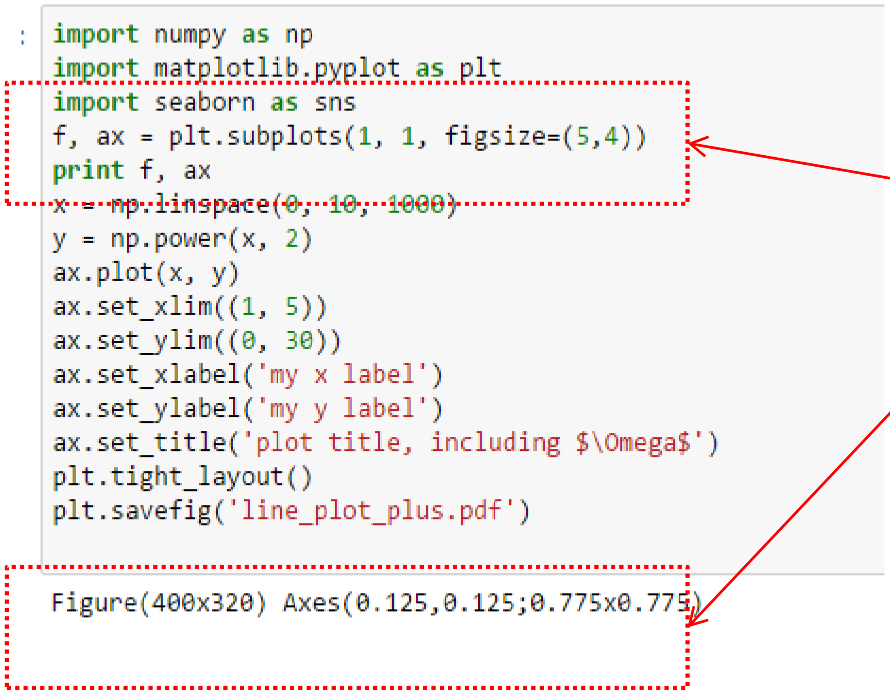
```
import numpy as np
import matplotlib.pyplot as plt

x = np.array([1, 2, 3, 4, 5])
y = np.array([1, 4, 9, 16, 25])
line1, line2 = plt.plot(x, y, '-', x,y*2,'--')
print line1
print line2

plt.setp(line1, color='r', linewidth=2.0)

plt.show()
```

```
Line2D(_line0)
Line2D(_line1)
```

Axes 객체 처리

Figure/Axes 객체 생성

객체를 생성



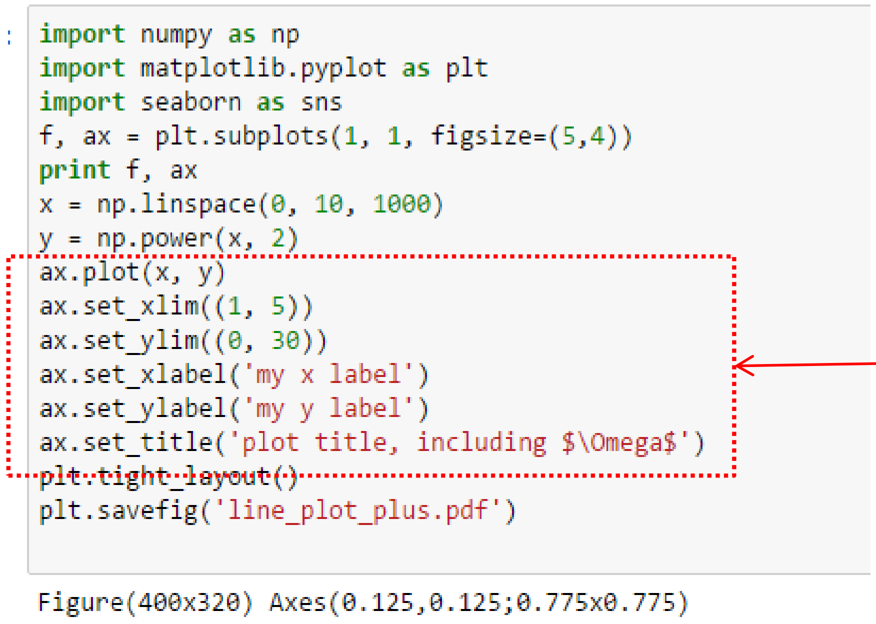
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
f, ax = plt.subplots(1, 1, figsize=(5,4))
print f, ax
x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
ax.plot(x, y)
ax.set_xlim((1, 5))
ax.set_ylim((0, 30))
ax.set_xlabel('my x label')
ax.set_ylabel('my y label')
ax.set_title('plot title, including $\Omega$')
plt.tight_layout()
plt.savefig('line_plot_plus.pdf')
```

```
Figure(400x320) Axes(0.125,0.125;0.775x0.775)
```

Axes 클래스의 인스턴스를 생성

Axes 내의 메소드 세팅

set_xlim, set_ylim, xlabel, ylabel, title을 세팅



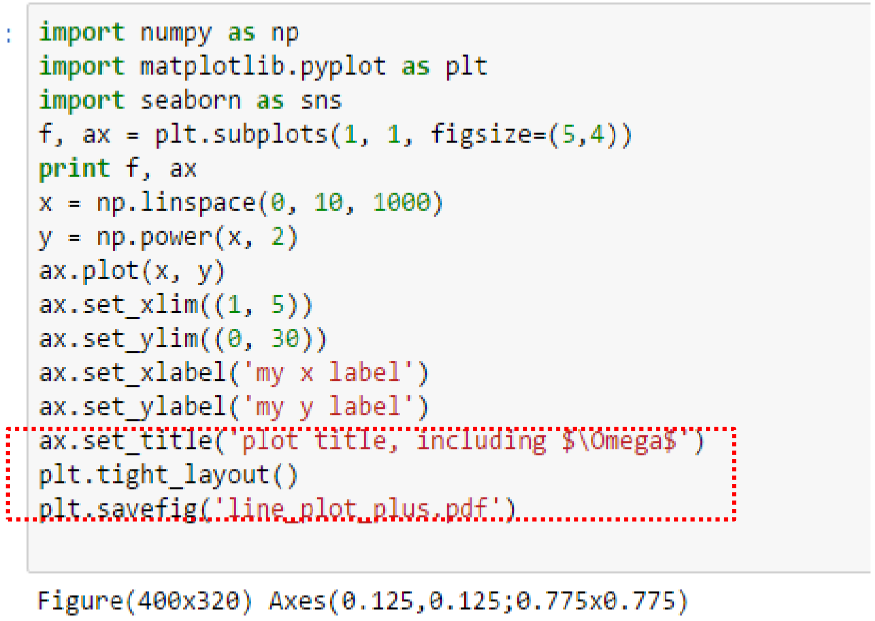
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
f, ax = plt.subplots(1, 1, figsize=(5,4))
print f, ax
x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
ax.plot(x, y)
ax.set_xlim((1, 5))
ax.set_ylim((0, 30))
ax.set_xlabel('my x label')
ax.set_ylabel('my y label')
ax.set_title('plot title, including $\Omega$')
plt.tight_layout()
plt.savefig('line_plot_plus.pdf')
```

```
Figure(400x320) Axes(0.125,0.125;0.775x0.775)
```

좌표 범위, label, 타이틀을 선언

tight_layout 처리

Text 객체 내에 지정된 것들을 재조정 처리

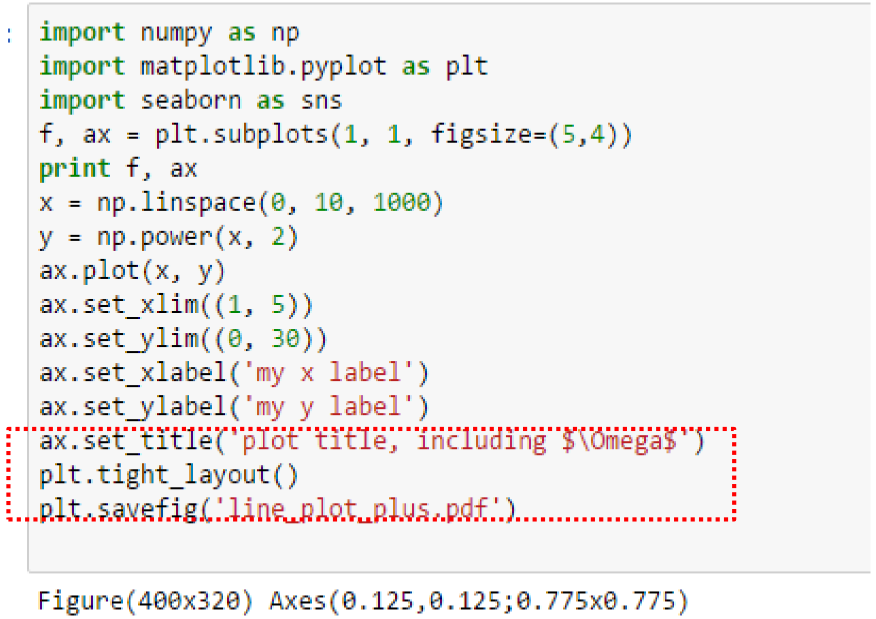


```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
f, ax = plt.subplots(1, 1, figsize=(5,4))
print f, ax
x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
ax.plot(x, y)
ax.set_xlim((1, 5))
ax.set_ylim((0, 30))
ax.set_xlabel('my x label')
ax.set_ylabel('my y label')
ax.set_title('plot title, including $\Omega$')
plt.tight_layout()
plt.savefig('line_plot_plus.pdf')
```

```
Figure(400x320) Axes(0.125,0.125;0.775x0.775)
```

tight_layout 처리

Text 객체 내에 지정된 것들을 재조정 처리



```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
f, ax = plt.subplots(1, 1, figsize=(5,4))
print f, ax
x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
ax.plot(x, y)
ax.set_xlim((1, 5))
ax.set_ylim((0, 30))
ax.set_xlabel('my x label')
ax.set_ylabel('my y label')
ax.set_title('plot title, including $\Omega$')
plt.tight_layout()
plt.savefig('line_plot_plus.pdf')
```

```
Figure(400x320) Axes(0.125,0.125;0.775x0.775)
```

MATPLOTLIB PYPLOT SCATTER 함수

Moon Yong Joon

점 그래프

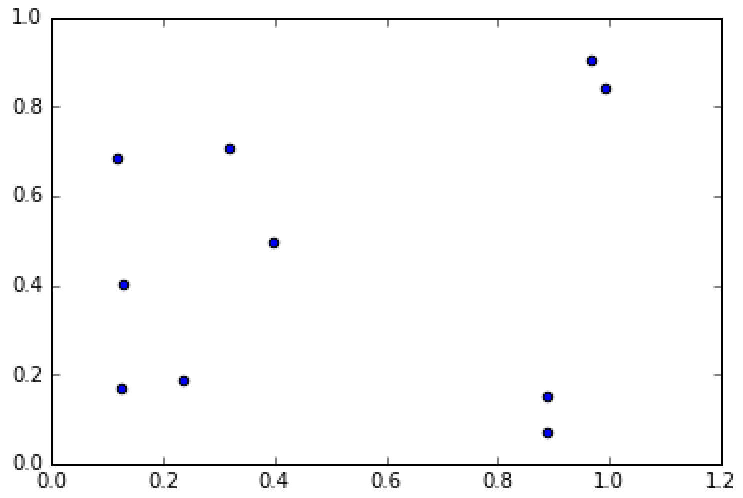
scatter 함수 : 분포점을 그리기

PathCollection object가 생기고 행 10과 열 2개의 데이터를 생성해서 분포점을 그리기

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

data = np.random.rand(10,2)
print data
print data.shape

plt.scatter(data[:,0],data[:,1])
plt.show()
```

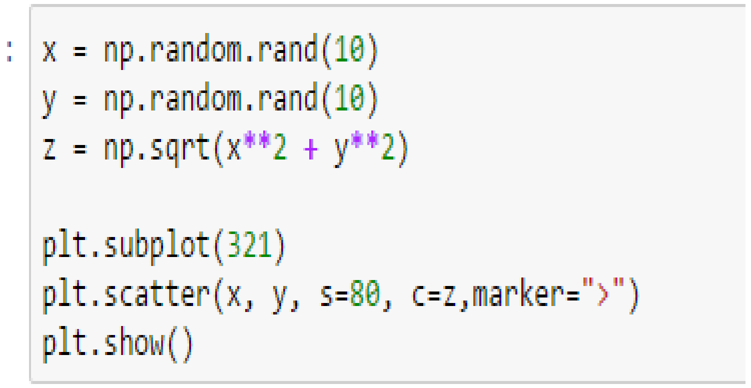
```
[[ 0.96772492  0.90626802]
 [ 0.12750724  0.40058642]
 [ 0.99235784  0.84277816]
 [ 0.12594414  0.16952064]
 [ 0.31897112  0.7056778 ]
 [ 0.88812603  0.15329952]
 [ 0.11697672  0.68375467]
 [ 0.88661934  0.06875145]
 [ 0.39632073  0.49774401]
 [ 0.23440602  0.18919622]]
(10, 2)
```

```
<matplotlib.collections.PathCollection object at 0x7f7d23cfc710>
```

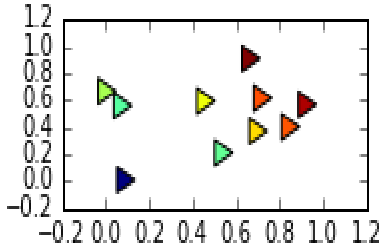


scatter 함수 : 모양과 색 입히기

s는 크기, c는 색상, marker는 삼각형



```
x = np.random.rand(10)
y = np.random.rand(10)
z = np.sqrt(x**2 + y**2)

plt.subplot(321)
plt.scatter(x, y, s=80, c=z,marker=">")
plt.show()
```



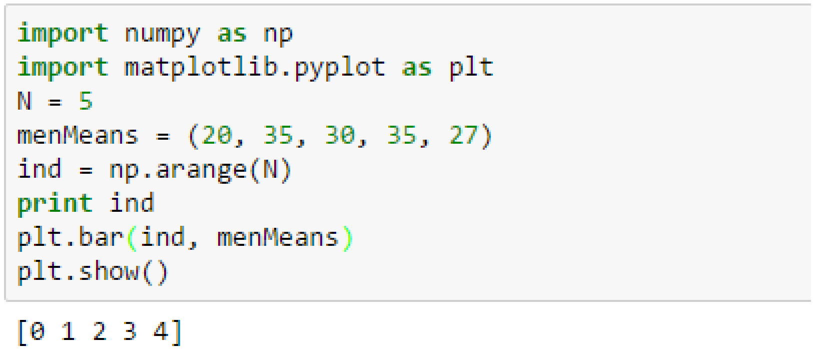
MATPLOTLIB PYPLOT BAR 함수

Moon Yong Joon

막대 그래프

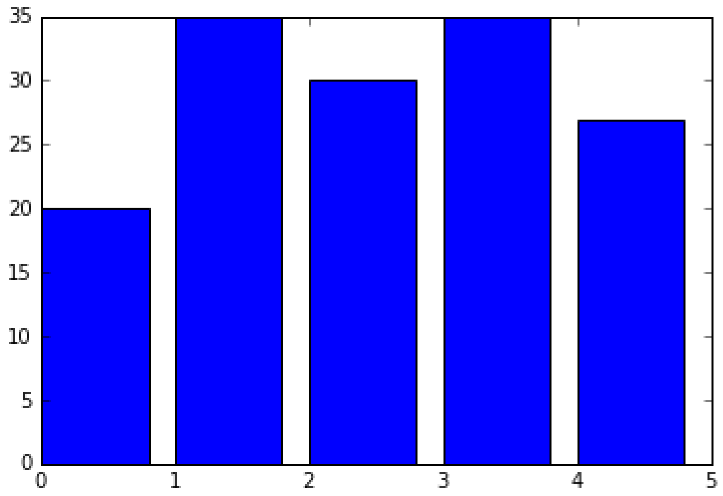
bar함수 : 기본

bar함수는 폭을 0.8, 파란색 막대가 기본으로 처리



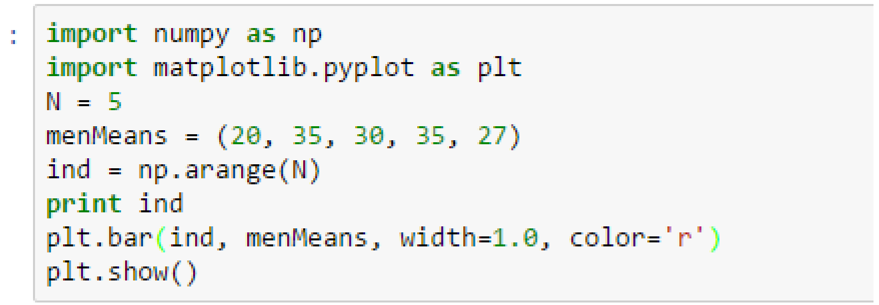
```
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans)
plt.show()
```

```
[0 1 2 3 4]
```



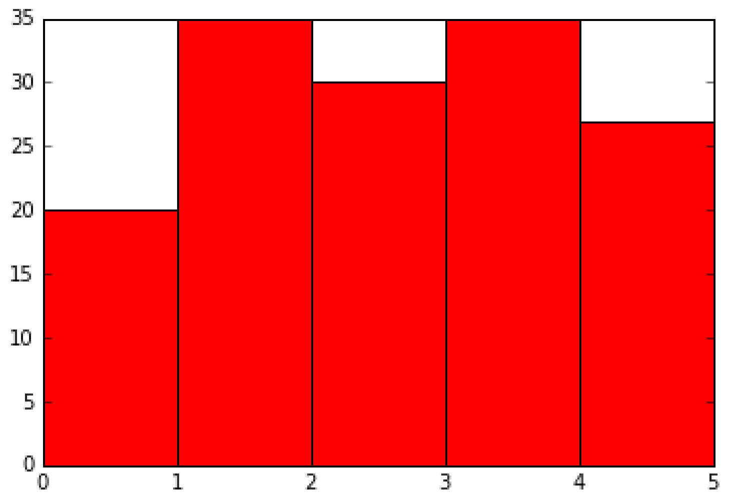
bar함수 : 폭 늘리기

bar함수는 위치와 값을 막대그래프로 표시



```
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans, width=1.0, color='r')
plt.show()
```

```
[0 1 2 3 4]
```

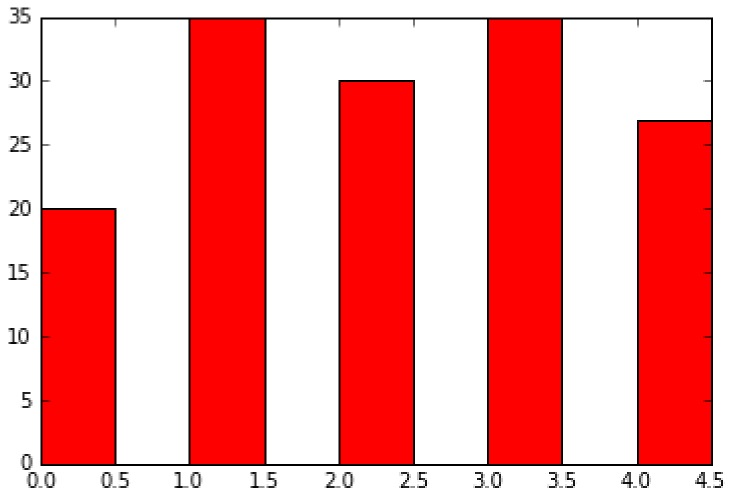


bar함수 : 폭 줄이기

bar함수는 막대 그래프의 폭을 0.5로 처리

```python
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans, width=0.5, color='r')
plt.show()
```

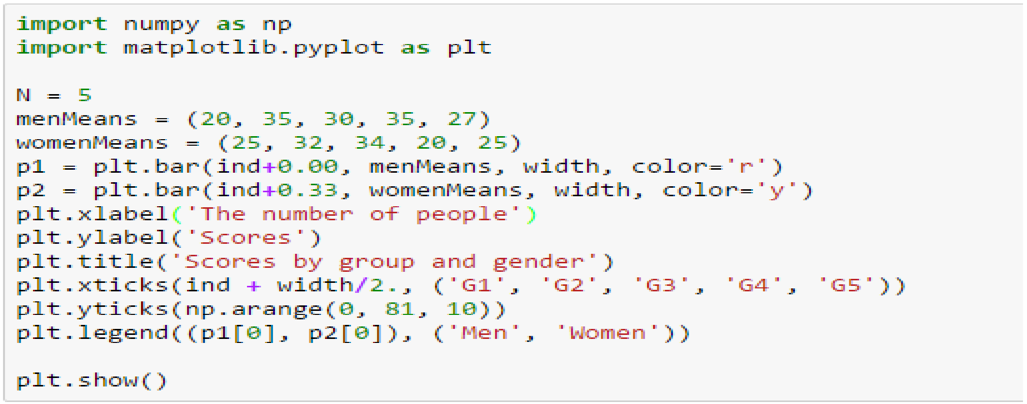
```
[0 1 2 3 4]
```



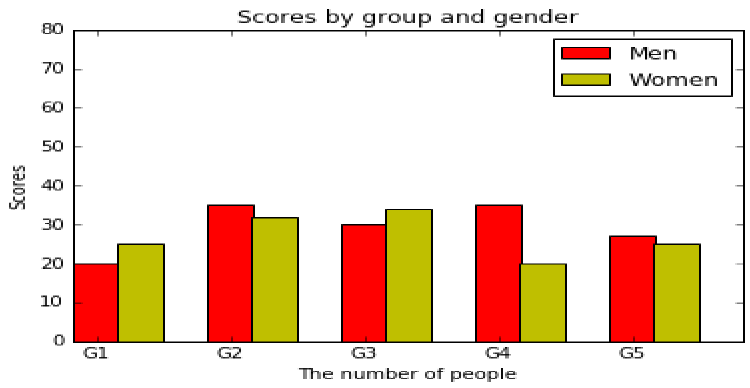
다중 막대그래프

bar함수 : 다중 막대그래프

bar함수는 막대 그래프의 폭을 0.33로 처리해 이중 막대 그래프



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

N = 5
menMeans = (20, 35, 30, 35, 27)
womenMeans = (25, 32, 34, 20, 25)
p1 = plt.bar(ind+0.00, menMeans, width, color='r')
p2 = plt.bar(ind+0.33, womenMeans, width, color='y')
plt.xlabel('The number of people')
plt.ylabel('Scores')
plt.title('Scores by group and gender')
plt.xticks(ind + width/2., ('G1', 'G2', 'G3', 'G4', 'G5'))
plt.yticks(np.arange(0, 81, 10))
plt.legend((p1[0], p2[0]), ('Men', 'Women'))

plt.show()
```



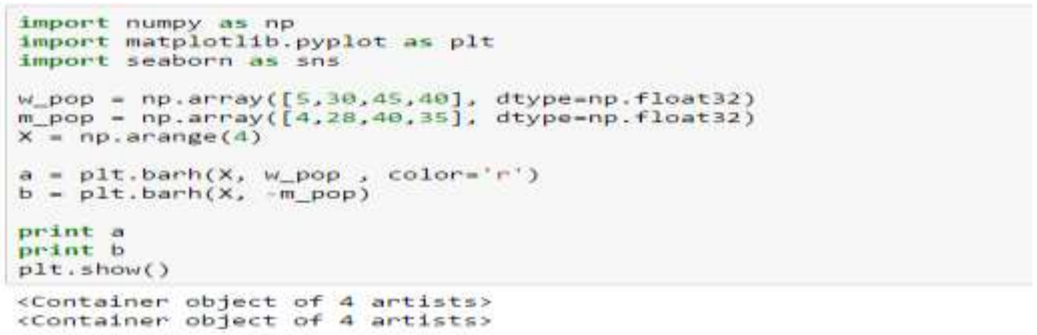
MATPLOTLIB PYPLOT BARH 함수

Moon Yong Joon

수평방향 막대 그래프

barh함수 : 수평 막대그래프

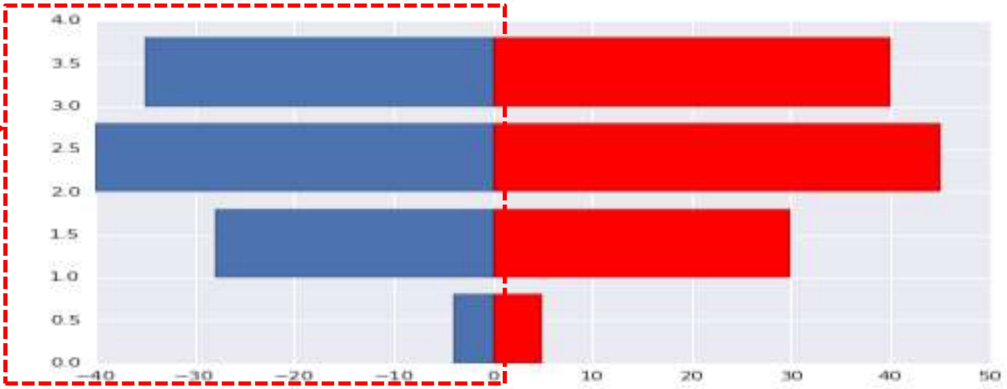
수평 막대그래프를 그리기 위해서는 반대방향의 데이터 m_pop 앞에 minus 부호(-)를 부여해야 함



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

w_pop = np.array([5,30,45,40], dtype=np.float32)
m_pop = np.array([4,28,40,35], dtype=np.float32)
x = np.arange(4)

a = plt.barh(x, w_pop , color='r')
b = plt.barh(x, -m_pop)

print a
print b
plt.show()
```

```
<Container object of 4 artists>
<Container object of 4 artists>
```

-m_pop 을 표시



MATPLOTLIB PYPLOT PIE 함수

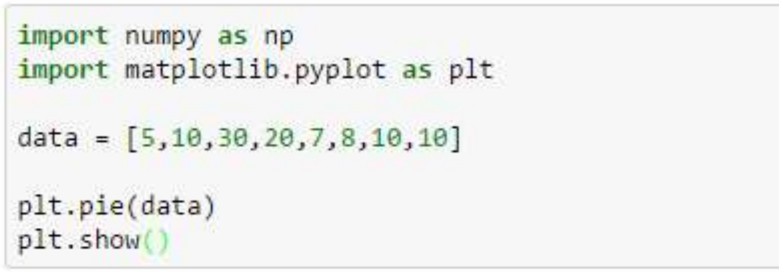
Moon Yong Joon

원 그래프

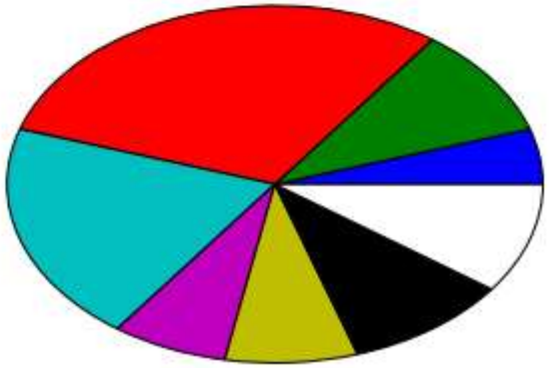
pie함수 :

데이터를 받아 원을 그래프 표시

색상 기본 순서 colors=('b', 'g', 'r', 'c', 'm', 'y', 'k', 'w')



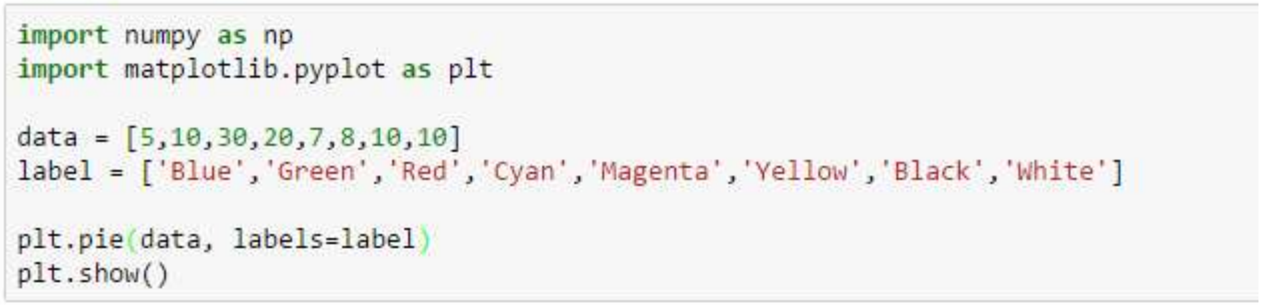
```
import numpy as np
import matplotlib.pyplot as plt

data = [5,10,30,20,7,8,10,10]

plt.pie(data)
plt.show()
```



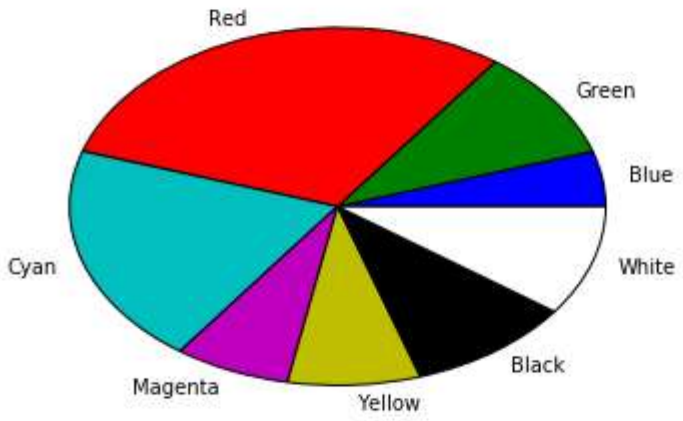
character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

pie함수 : labels 붙이기

데이터와 labels를 받아 원을 그래프 표시



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

data = [5,10,30,20,7,8,10,10]
label = ['Blue','Green','Red','Cyan','Magenta','Yellow','Black','White']

plt.pie(data, labels=label)
plt.show()
```



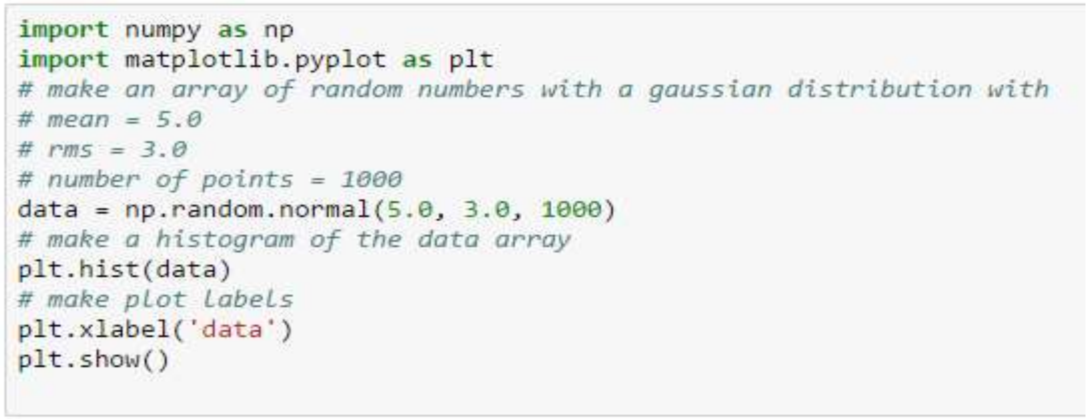
MATPLOTLIB PYPLOT HISTO 함수

Moon Yong Joon

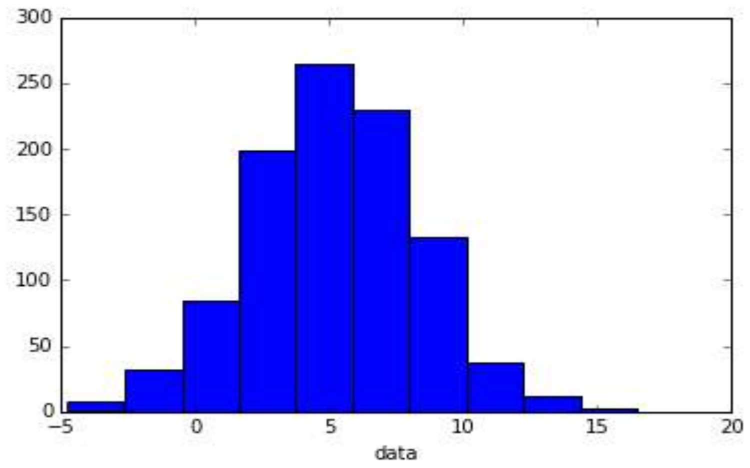
히스토그램 그래프

hist함수 : 기본

xlabel을 표시한 히스토그램 그리기

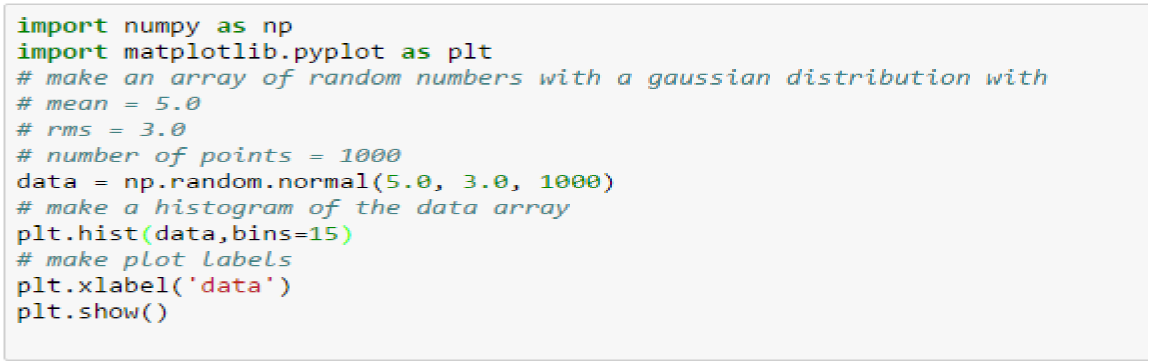


```
import numpy as np
import matplotlib.pyplot as plt
# make an array of random numbers with a gaussian distribution with
# mean = 5.0
# rms = 3.0
# number of points = 1000
data = np.random.normal(5.0, 3.0, 1000)
# make a histogram of the data array
plt.hist(data)
# make plot labels
plt.xlabel('data')
plt.show()
```

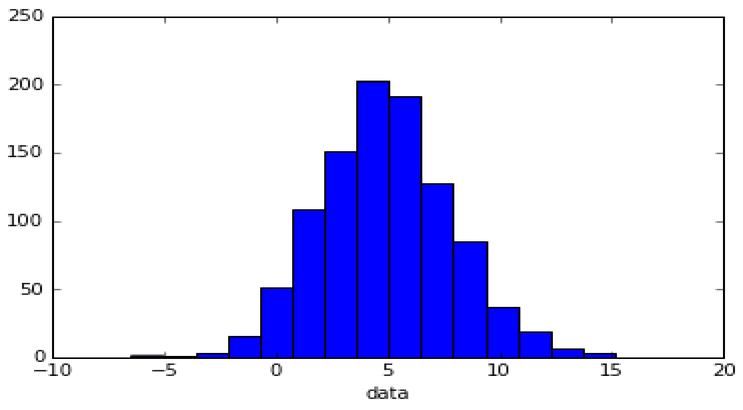


hist함수 : 범주 나누기 1

데이터를 받아 15개의 범주로 나눠 그래프를 표시

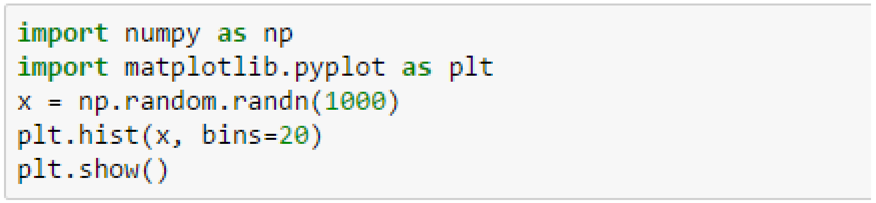


```
import numpy as np
import matplotlib.pyplot as plt
# make an array of random numbers with a gaussian distribution with
# mean = 5.0
# rms = 3.0
# number of points = 1000
data = np.random.normal(5.0, 3.0, 1000)
# make a histogram of the data array
plt.hist(data,bins=15)
# make plot labels
plt.xlabel('data')
plt.show()
```

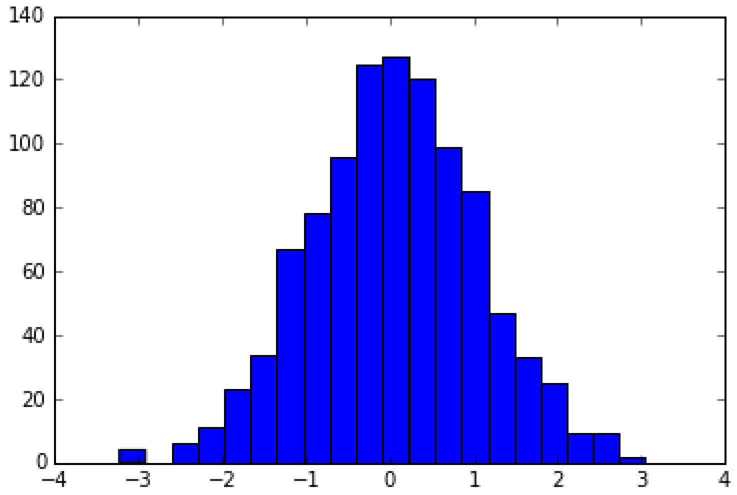


hist함수 : 범주 나누기 2

데이터를 받아 20개의 범주로 나눠 그래프를 표시

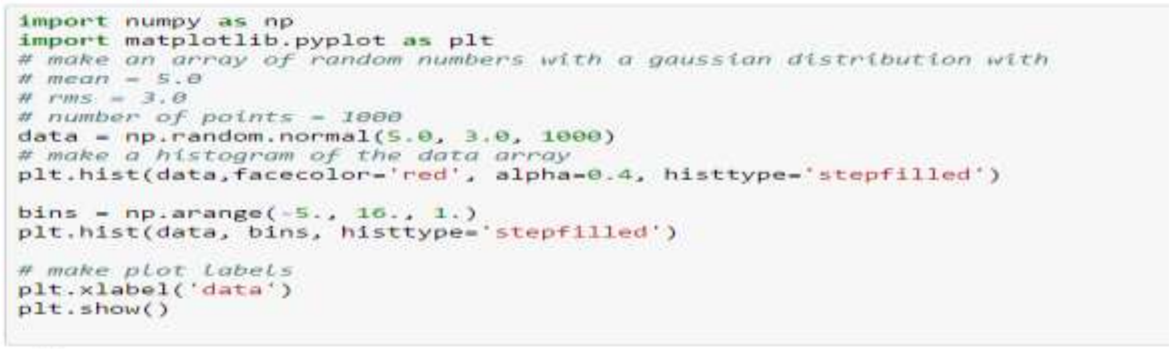


```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.randn(1000)
plt.hist(x, bins=20)
plt.show()
```

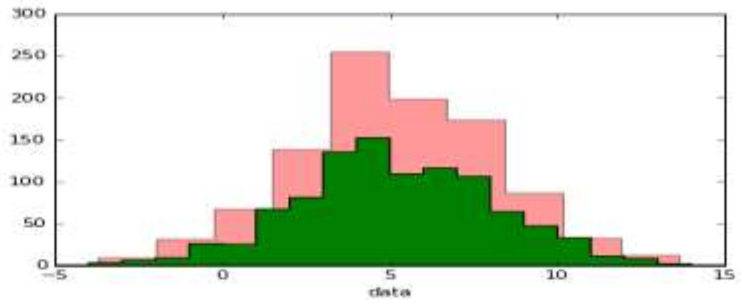


hist함수 : 내부에 그리기

그래프 내에 범주를 재정의해서 그래프를 그리기

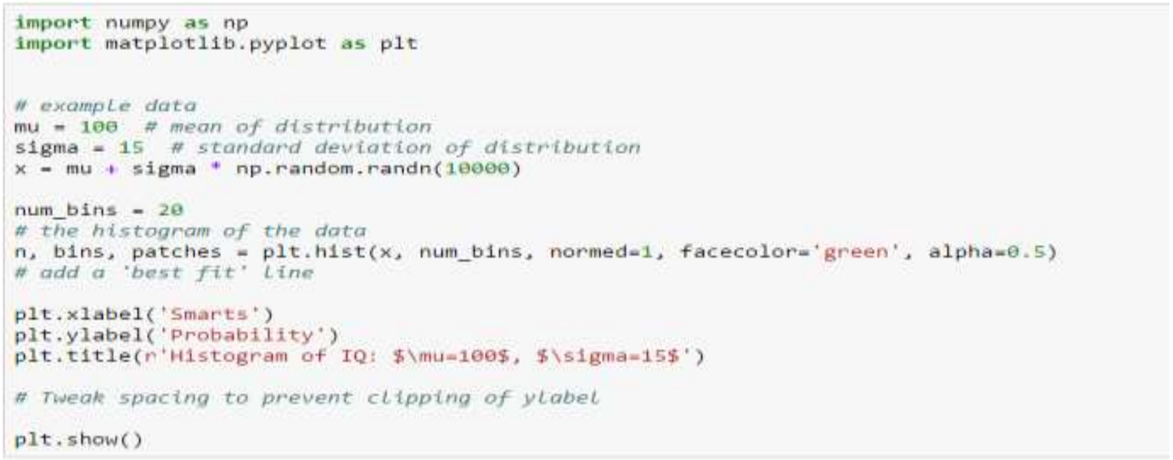


```
import numpy as np
import matplotlib.pyplot as plt
# make an array of random numbers with a gaussian distribution with
# mean = 5.0
# rms = 3.0
# number of points = 1000
data = np.random.normal(5.0, 3.0, 1000)
# make a histogram of the data array
plt.hist(data,facecolor='red', alpha=0.4, histtype='stepfilled')

bins = np.arange(-5., 16., 1.)
plt.hist(data, bins, histtype='stepfilled')

# make plot labels
plt.xlabel('data')
plt.show()
```

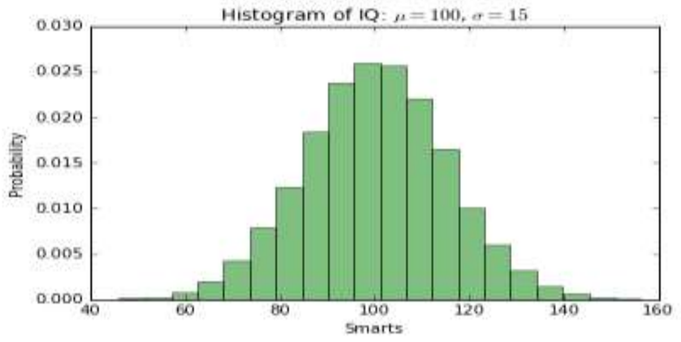


hist함수 :파라미터

normed를 사용하면 히스토그램 합이 1, facecolor는 색깔, alpha는 투명도 표시

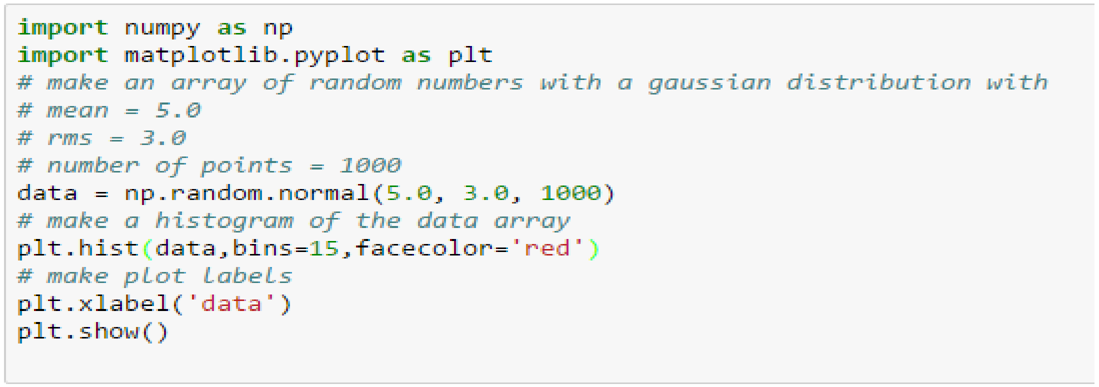


```
import numpy as np
import matplotlib.pyplot as plt


# example data
mu = 100  # mean of distribution
sigma = 15  # standard deviation of distribution
x = mu + sigma * np.random.randn(10000)

num_bins = 20
# the histogram of the data
n, bins, patches = plt.hist(x, num_bins, normed=1, facecolor='green', alpha=0.5)
# add a 'best fit' line

plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title(r'Histogram of IQ: $\mu=100$, $\sigma=15$')

# Tweak spacing to prevent clipping of ylabel

plt.show()
```

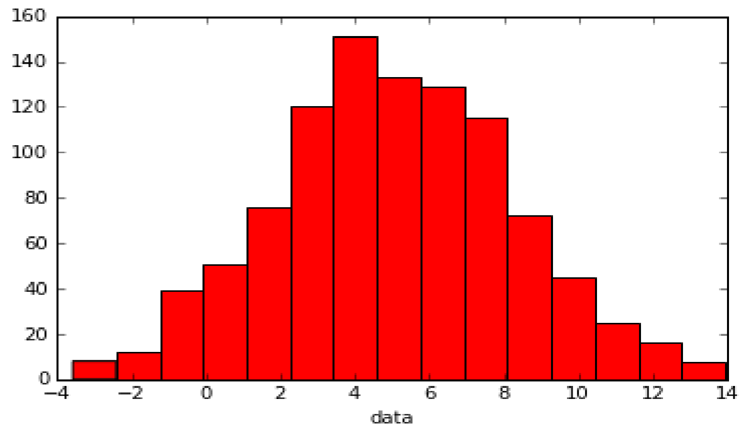


hist함수 : 색깔 바꾸기

facecolor에 red를 주고 색깔을 변경하기

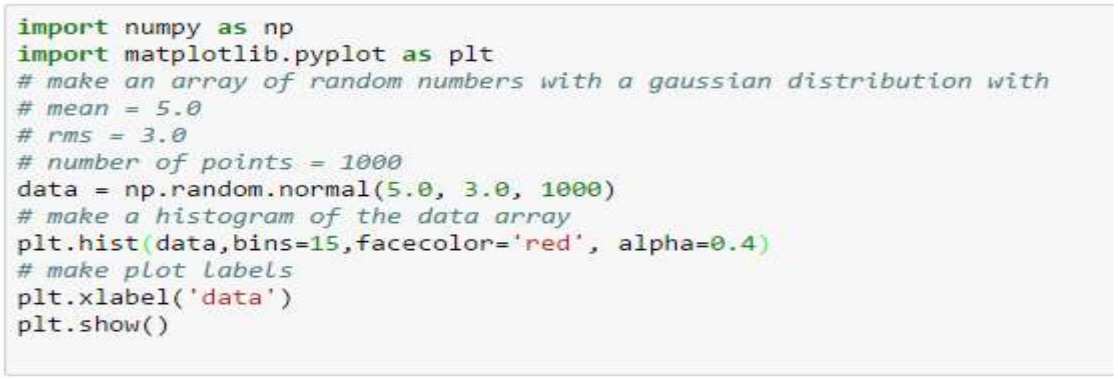


```
import numpy as np
import matplotlib.pyplot as plt
# make an array of random numbers with a gaussian distribution with
# mean = 5.0
# rms = 3.0
# number of points = 1000
data = np.random.normal(5.0, 3.0, 1000)
# make a histogram of the data array
plt.hist(data,bins=15,facecolor='red')
# make plot labels
plt.xlabel('data')
plt.show()
```

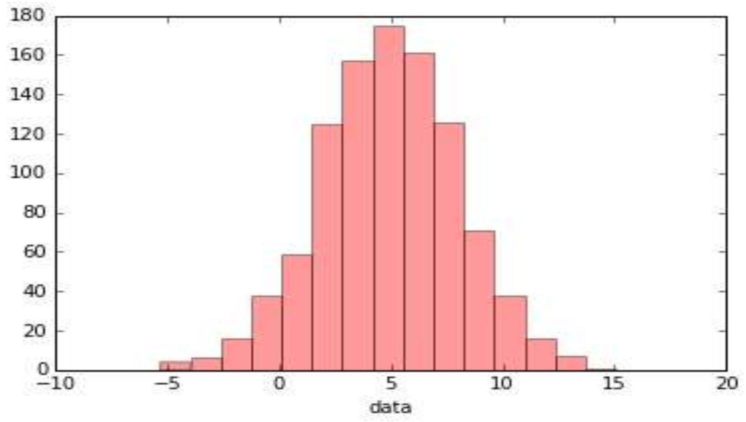


hist함수 : 투명도 조정

alpha에 0.4를 주고 색깔에 대한 투명도를 조정하기

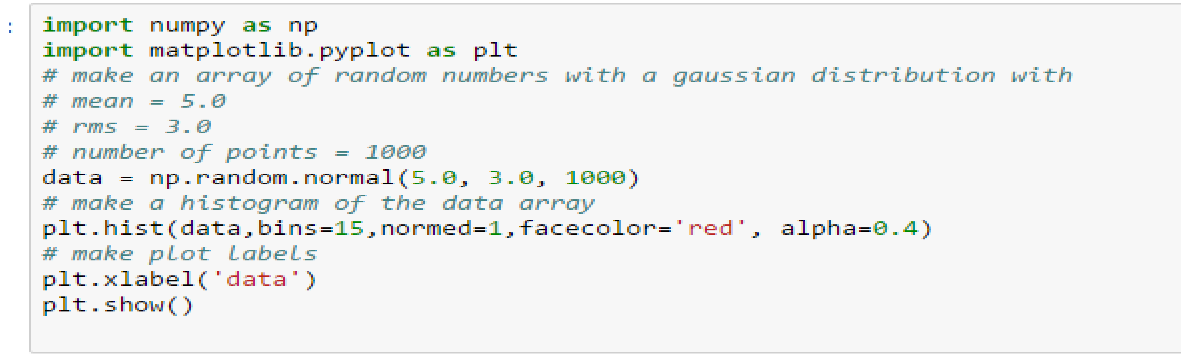


```python
import numpy as np
import matplotlib.pyplot as plt
# make an array of random numbers with a gaussian distribution with
# mean = 5.0
# rms = 3.0
# number of points = 1000
data = np.random.normal(5.0, 3.0, 1000)
# make a histogram of the data array
plt.hist(data,bins=15,facecolor='red', alpha=0.4)
# make plot labels
plt.xlabel('data')
plt.show()
```

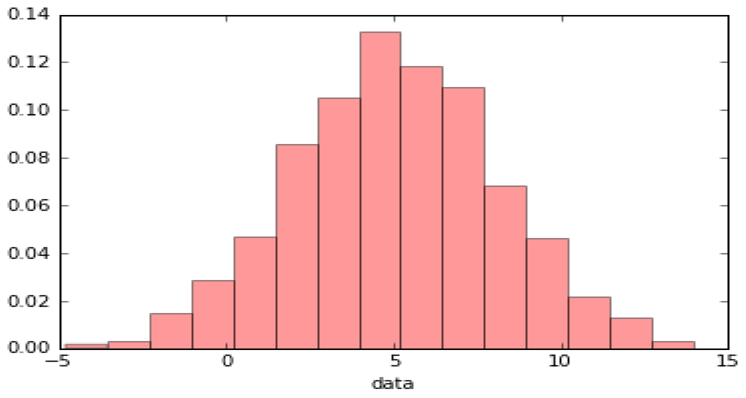


hist함수 : 전체 비율값 1로 조정

normed에 1를 주면 앞의 전체 비율이 합이 1로 처리

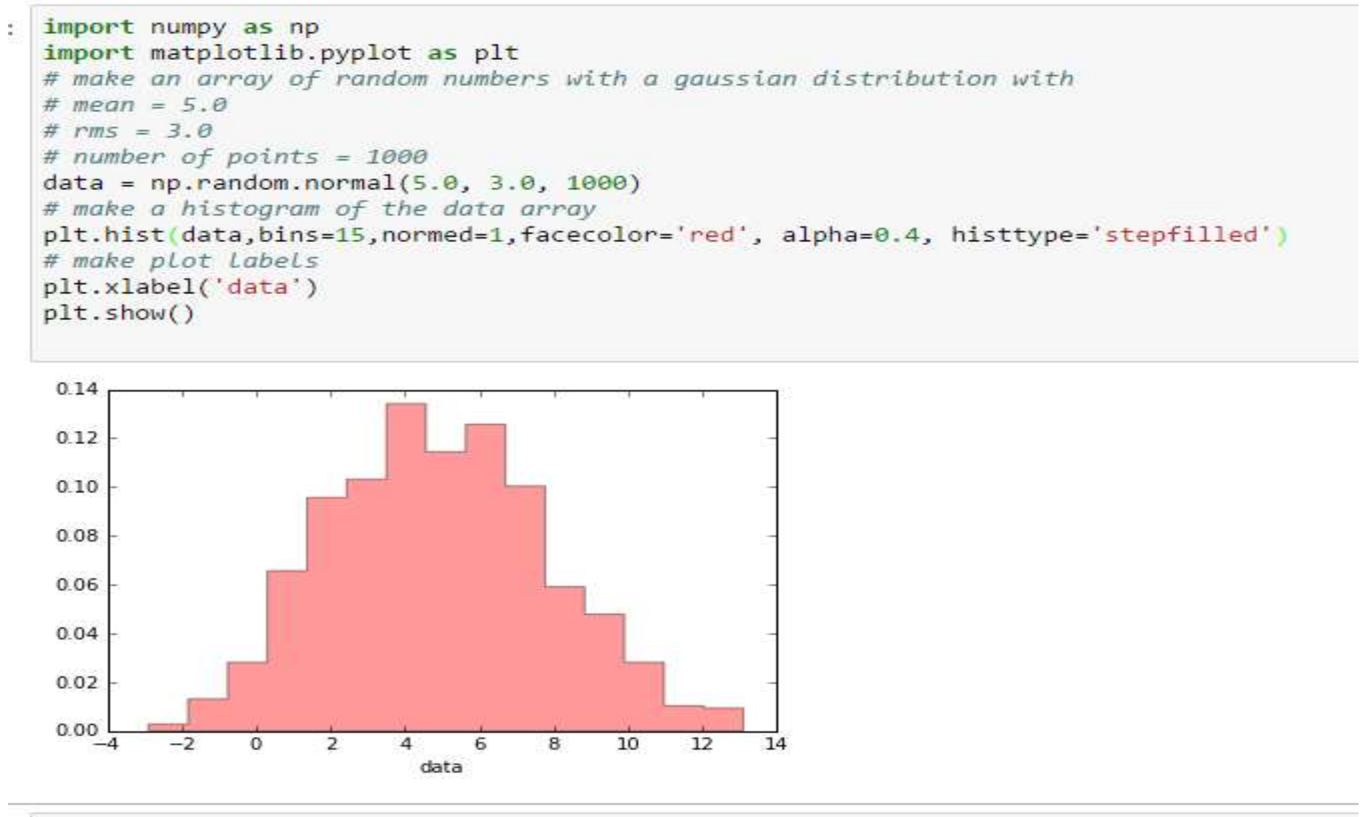


```
import numpy as np
import matplotlib.pyplot as plt
# make an array of random numbers with a gaussian distribution with
# mean = 5.0
# rms = 3.0
# number of points = 1000
data = np.random.normal(5.0, 3.0, 1000)
# make a histogram of the data array
plt.hist(data,bins=15,normed=1,facecolor='red', alpha=0.4)
# make plot labels
plt.xlabel('data')
plt.show()
```



hist함수 : histtype

histtype에 stepfilled를 주면 경계선이 없어짐



```
import numpy as np
import matplotlib.pyplot as plt
# make an array of random numbers with a gaussian distribution with
# mean = 5.0
# rms = 3.0
# number of points = 1000
data = np.random.normal(5.0, 3.0, 1000)
# make a histogram of the data array
plt.hist(data,bins=15,normed=1,facecolor='red', alpha=0.4, histtype='stepfilled')
# make plot labels
plt.xlabel('data')
plt.show()
```

Axes 객체 처리

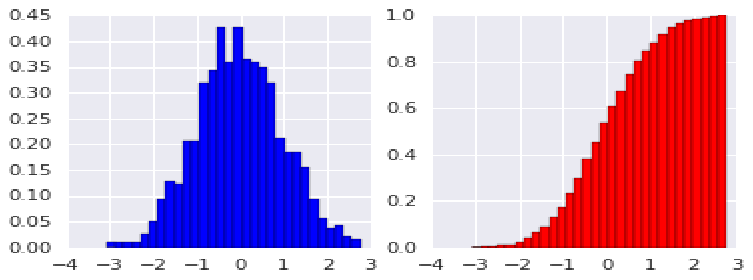
Axes.hist 메소드

hist 메소드를 이용해서 처리

Cumulative는 누적 분포를 나타내는 그래프를 추가로 그리기 위한 파라미터

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data = np. random. randn(1000)
f , (ax1, ax2) = plt.subplots (1, 2, figsize =(6,3))
print ax1
print ax2
# histogram (pdf)
ax1.hist(data , bins=30, normed=True, color='b')
# empirical cdf
ax2.hist(data , bins=30, normed=True, color='r',cumulative=True)
plt.show()
```

```
Axes(0.125,0.125;0.352273x0.775)
Axes(0.547727,0.125;0.352273x0.775)
```



MATPLOTLIB PYPLOT BOXPLOT함수

Moon Yong Joon

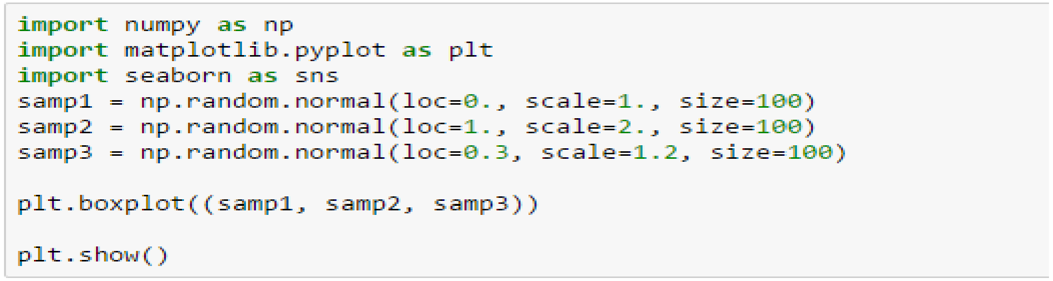
box 그래프

Axes.boxplot 메소드

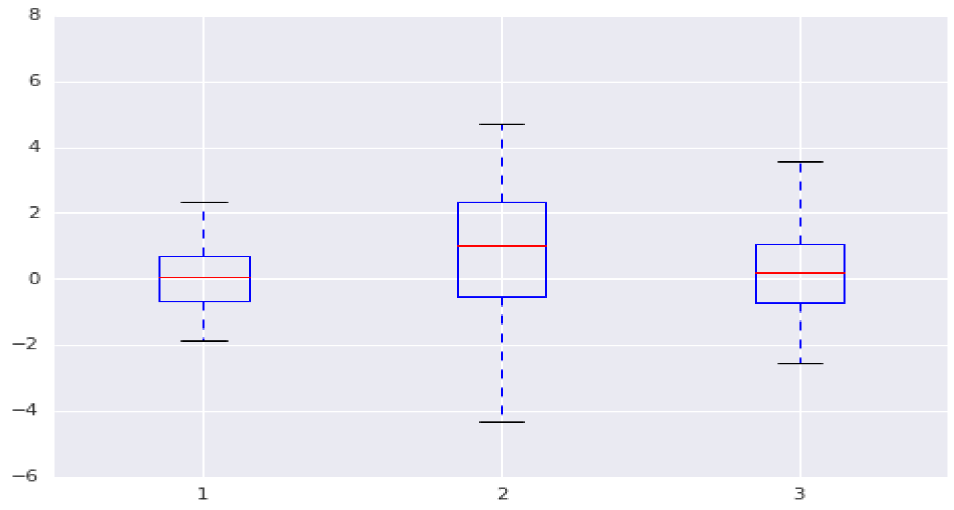
boxplot 메소드를 이용해서 처리

boxplot 함수

boxplot 함수를 이용해서 처리



```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
samp1 = np.random.normal(loc=0., scale=1., size=100)
samp2 = np.random.normal(loc=1., scale=2., size=100)
samp3 = np.random.normal(loc=0.3, scale=1.2, size=100)

plt.boxplot((samp1, samp2, samp3))

plt.show()
```

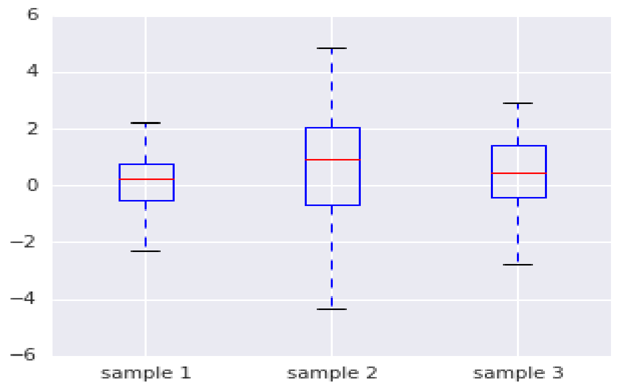


Axes 객체 처리

Axes.boxplot 메소드

boxplot 메소드를 이용해서 처리

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
samp1 = np.random.normal(loc=0., scale=1., size=100)
samp2 = np.random.normal(loc=1., scale=2., size=100)
samp3 = np.random.normal(loc=0.3, scale=1.2, size=100)
f, ax = plt.subplots(1, 1, figsize=(5,4))
ax.boxplot((samp1, samp2, samp3))
ax.set_xticklabels(['sample 1', 'sample 2', 'sample 3'])
plt.show()
```



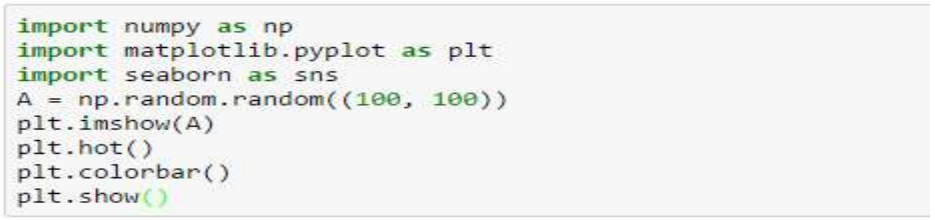
MATPLOTLIB PYPLOT IMAGE PLOT함수

Moon Yong Joon

이미지 그래프

isshow() 함수

imshow() 함수를 이용해서 이미지 출력
colorbar함수를 이용옆에 옆에 colorbar를 출력



```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
A = np.random.random((100, 100))
plt.imshow(A)
plt.hot()
plt.colorbar()
plt.show()
```

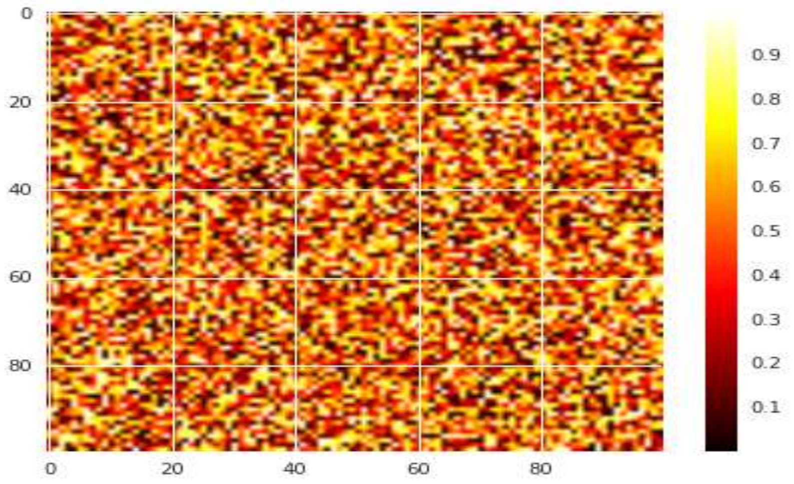
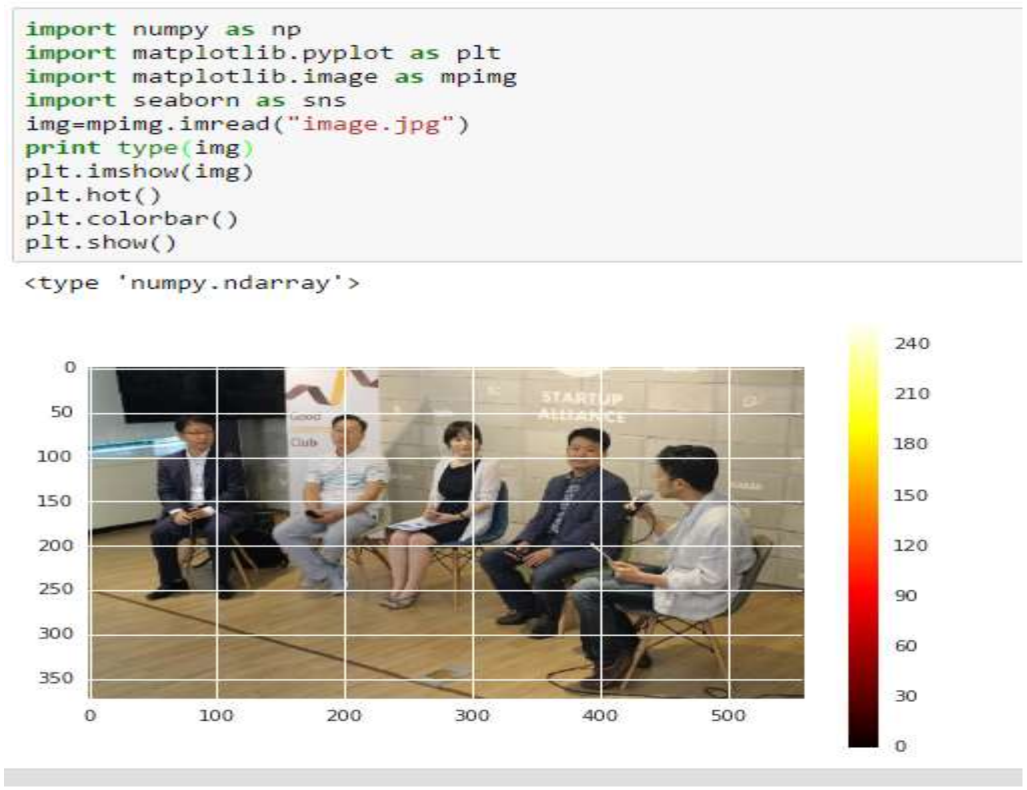


image.read 함수

이미지 파일을 읽고 이를 ndarray로 전환해서 imshow함수로 그래프 출력

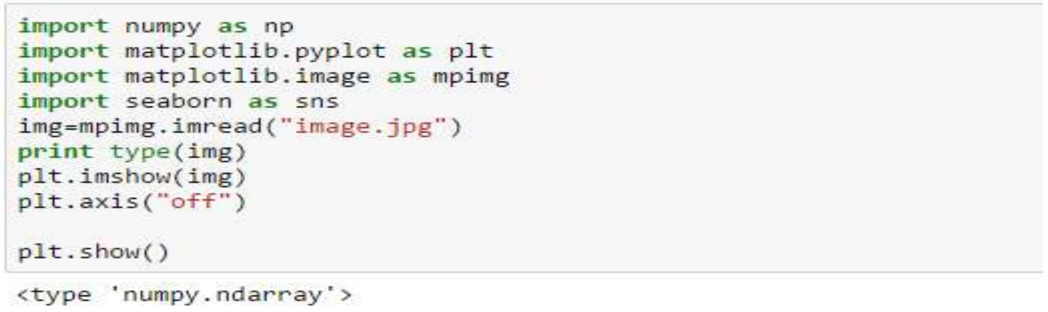


```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
img=mpimg.imread("image.jpg")
print type(img)
plt.imshow(img)
plt.hot()
plt.colorbar()
plt.show()
```

```
<type 'numpy.ndarray'>
```

이미지 처리시 좌표축 제거하기

axis('off')를 이용해서 이미지만 출력



```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
img=mpimg.imread("image.jpg")
print type(img)
plt.imshow(img)
plt.axis("off")

plt.show()
```

```
<type 'numpy.ndarray'>
```

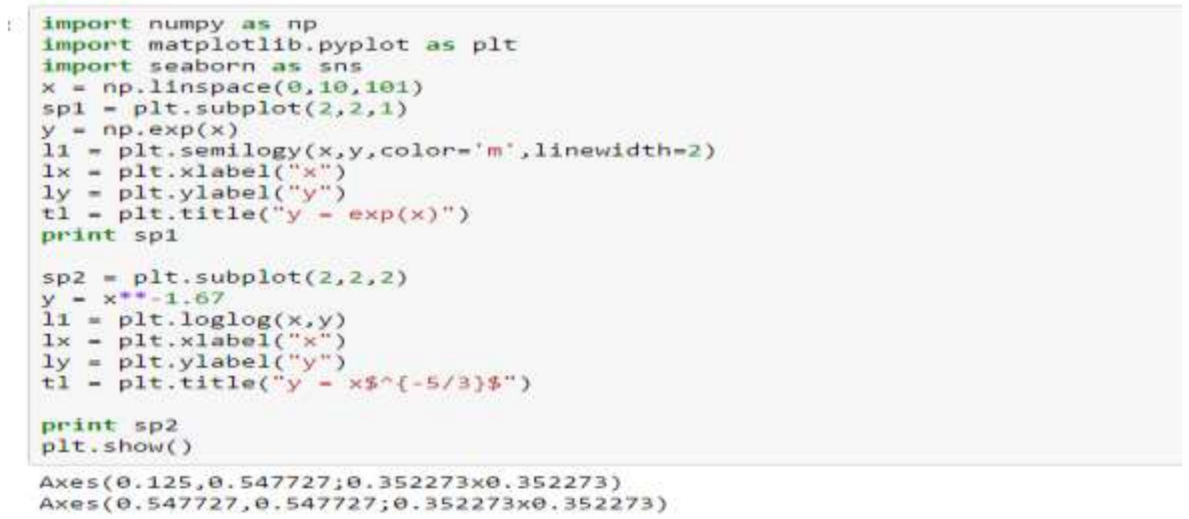
MATPLOTLIB PYPLOT LOGPLOT 함수

Moon Yong Joon

log 그래프

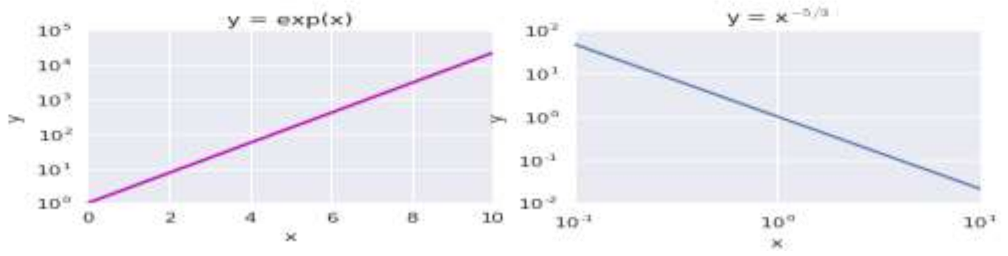
subplot 사용시 2개 Axes 생성

Axes 객체를 2개 생성해서 그래프를 2개로 분리



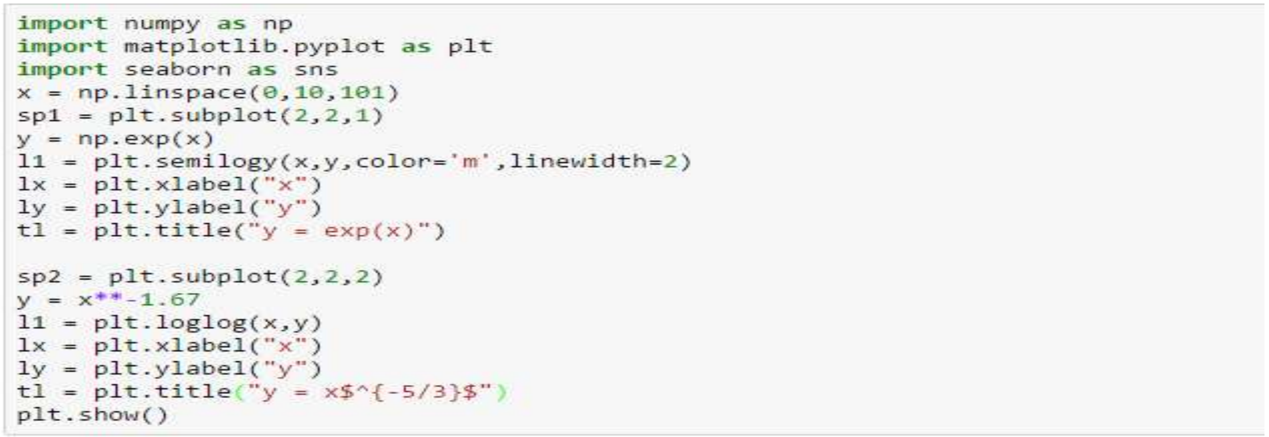
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
x = np.linspace(0,10,101)
sp1 = plt.subplot(2,2,1)
y = np.exp(x)
l1 = plt.semilogy(x,y,color='m',linewidth=2)
lx = plt.xlabel("x")
ly = plt.ylabel("y")
tl = plt.title("y = exp(x)")
print sp1

sp2 = plt.subplot(2,2,2)
y = x**-1.67
l1 = plt.loglog(x,y)
lx = plt.xlabel("x")
ly = plt.ylabel("y")
tl = plt.title("y = x$^{-5/3}$")

print sp2
plt.show()
```

```
Axes(0.125,0.547727;0.352273x0.352273)
Axes(0.547727,0.547727;0.352273x0.352273)
```

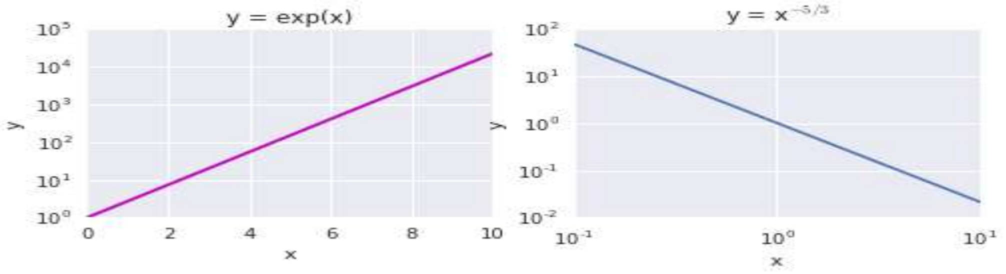


semilogy/loglog 함수

log를 처리한 결과를 그래프로 표시



```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
x = np.linspace(0,10,101)
sp1 = plt.subplot(2,2,1)
y = np.exp(x)
l1 = plt.semilogy(x,y,color='m',linewidth=2)
lx = plt.xlabel("x")
ly = plt.ylabel("y")
tl = plt.title("y = exp(x)")

sp2 = plt.subplot(2,2,2)
y = x**-1.67
l1 = plt.loglog(x,y)
lx = plt.xlabel("x")
ly = plt.ylabel("y")
tl = plt.title("y = x$^{-5/3}$")
plt.show()
```



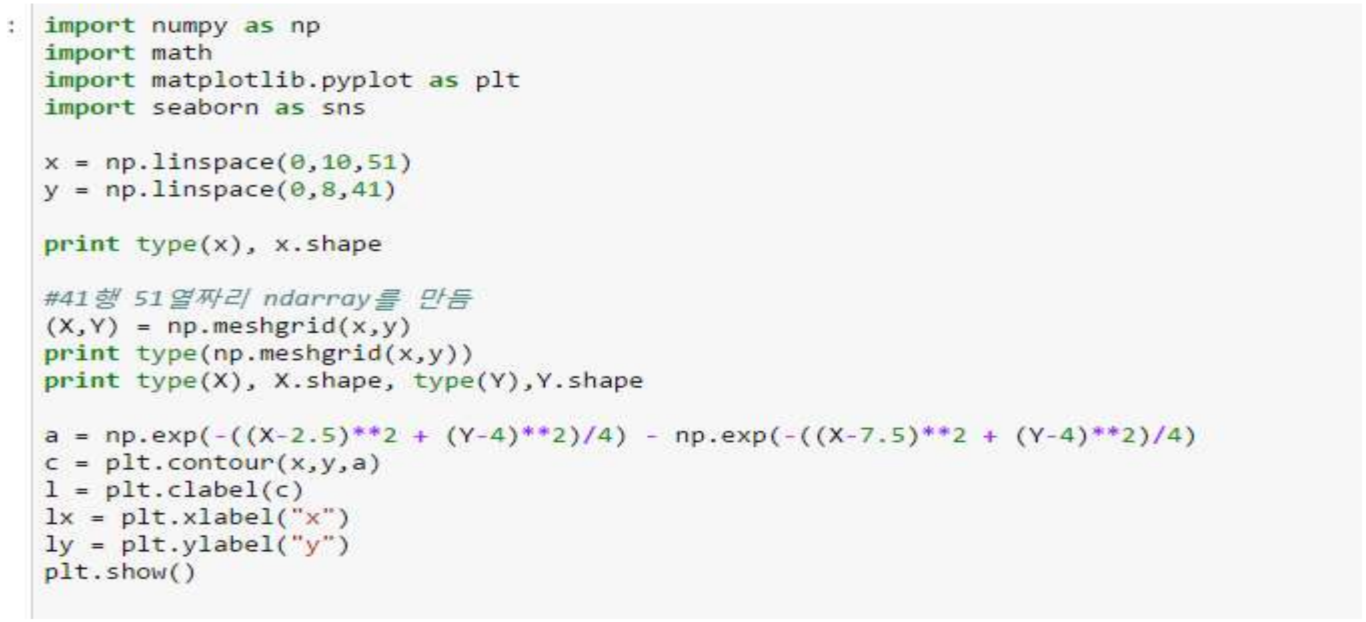
MATPLOTLIB **TWO-DIMENSIONAL PLOTS 함수**

Moon Yong Joon

Contour plots

데이터 구조 이해하기

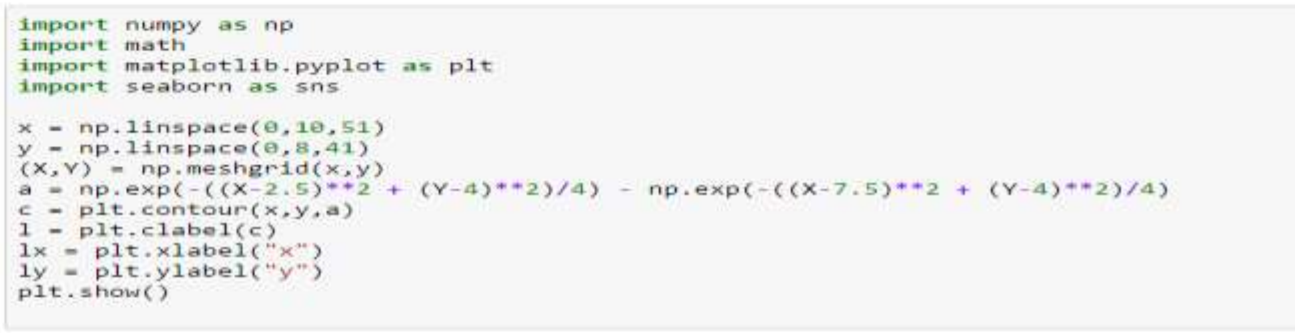
meshgrid 함수를 이용해서 2개의 같은 차원의 ndarray 를생성



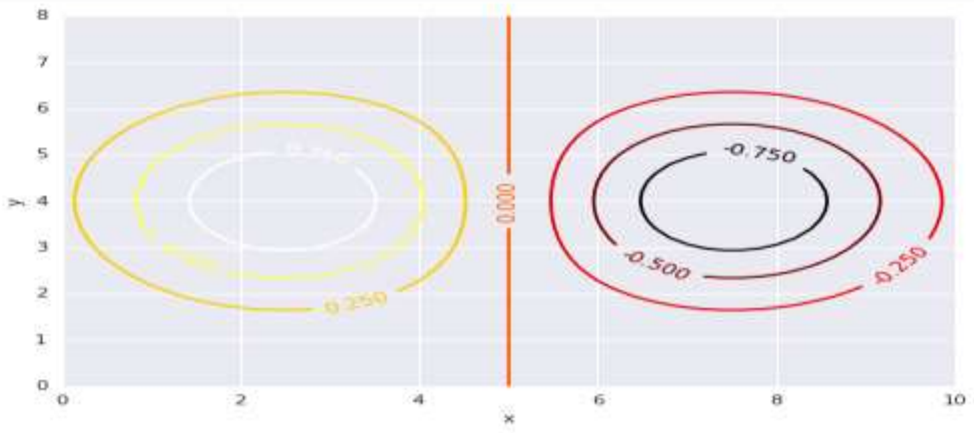
```
import numpy as np
import math
import matplotlib.pyplot as plt
import seaborn as sns

x = np.linspace(0,10,51)
y = np.linspace(0,8,41)

print type(x), x.shape

#41행 51열짜리 ndarray를 만듬
(X,Y) = np.meshgrid(x,y)
print type(np.meshgrid(x,y))
print type(X), X.shape, type(Y),Y.shape

a = np.exp(-((X-2.5)**2 + (Y-4)**2)/4) - np.exp(-((X-7.5)**2 + (Y-4)**2)/4)
c = plt.contour(x,y,a)
l = plt.clabel(c)
lx = plt.xlabel("x")
ly = plt.ylabel("y")
plt.show()
```

Contour plots

Contour plots을 이용해서 여러 원에 대해 그래기



```
import numpy as np
import math
import matplotlib.pyplot as plt
import seaborn as sns

x = np.linspace(0,10,51)
y = np.linspace(0,8,41)
(X,Y) = np.meshgrid(x,y)
a = np.exp(-((X-2.5)**2 + (Y-4)**2)/4) - np.exp(-((X-7.5)**2 + (Y-4)**2)/4)
c = plt.contour(x,y,a)
l = plt.clabel(c)
lx = plt.xlabel("x")
ly = plt.ylabel("y")
plt.show()
```



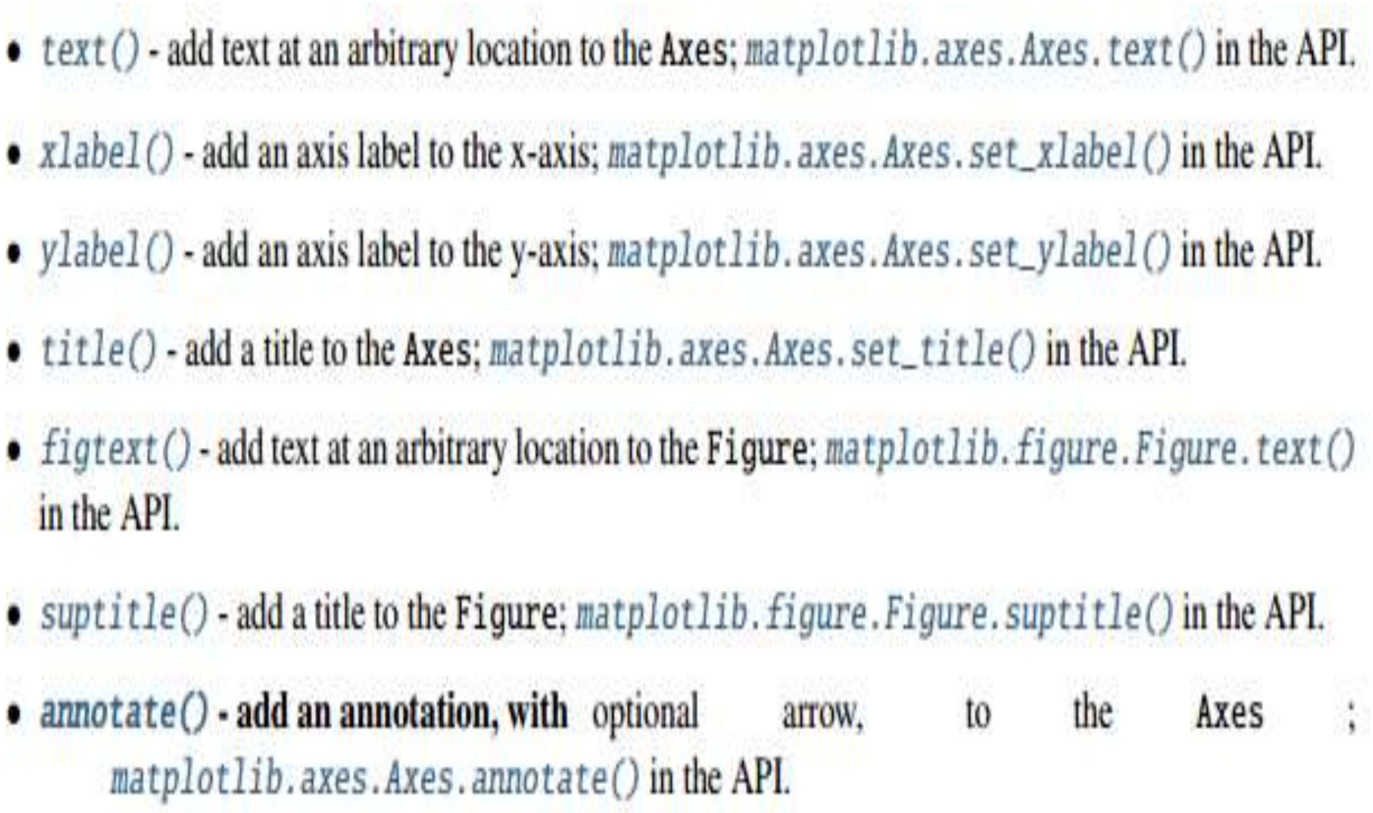
MATPLOTLIB 그래프 꾸미기 텍스트 처리

Moon Yong Joon

Basic text commands

Basic text commands

Basic text commands 함수들



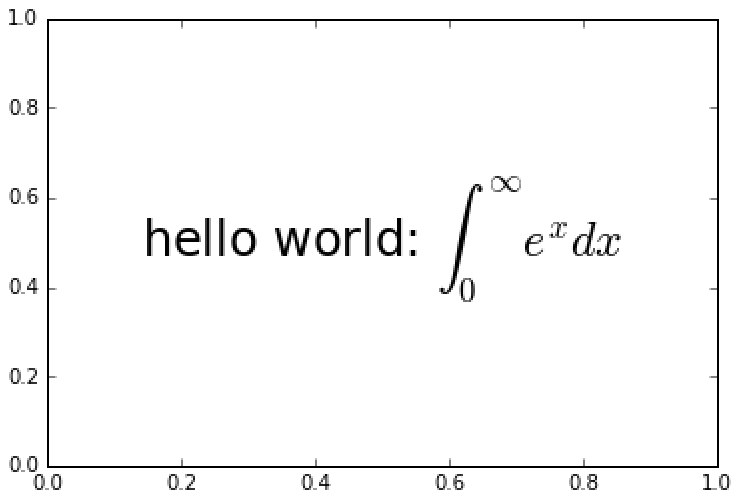
- text() - add text at an arbitrary location to the Axes; matplotlib.axes.Axes.text() in the API.
- xlabel() - add an axis label to the x-axis; matplotlib.axes.Axes.set_xlabel() in the API.
- ylabel() - add an axis label to the y-axis; matplotlib.axes.Axes.set_ylabel() in the API.
- title() - add a title to the Axes; matplotlib.axes.Axes.set_title() in the API.
- figtext() - add text at an arbitrary location to the Figure; matplotlib.figure.Figure.text() in the API.
- suptitle() - add a title to the Figure; matplotlib.figure.Figure.suptitle() in the API.
- annotate() - add an annotation, with optional arrow, to the Axes ; matplotlib.axes.Axes.annotate() in the API.

text

text 함수 : 기초

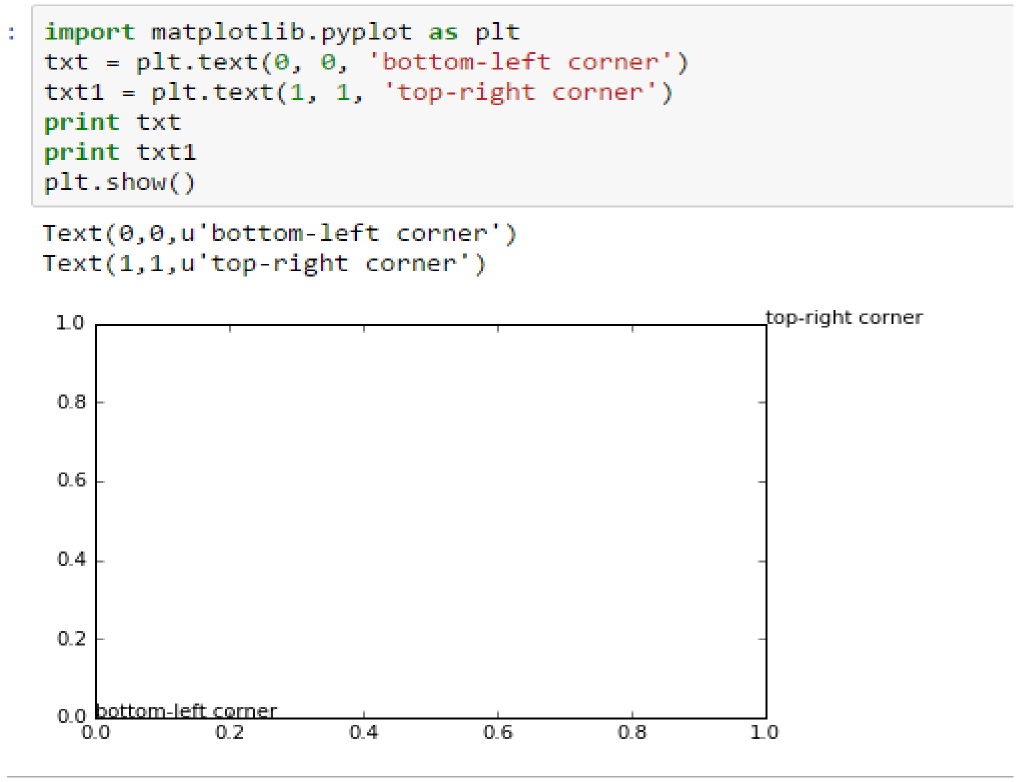
그래프 내에 특정 좌표에 문자열이 들어가도록 입력해서 표시

```
import numpy as np
import matplotlib.pyplot as plt
plt.text(0.5, 0.5, 'hello world: $\int_0^\infty e^x dx$', size=24, ha='center', va='center')
plt.show()
```



text 함수 : 좌표에 따른 표시

text함수는 Text 클래스의 객체를 생성하고 그 위치 값을 좌표로 해서 문자열을 출력함

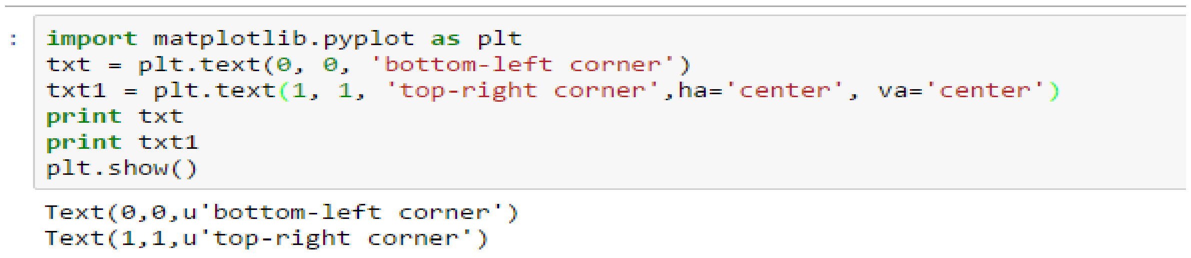


```
import matplotlib.pyplot as plt
txt = plt.text(0, 0, 'bottom-left corner')
txt1 = plt.text(1, 1, 'top-right corner')
print txt
print txt1
plt.show()
```

```
Text(0,0,u'bottom-left corner')
Text(1,1,u'top-right corner')
```

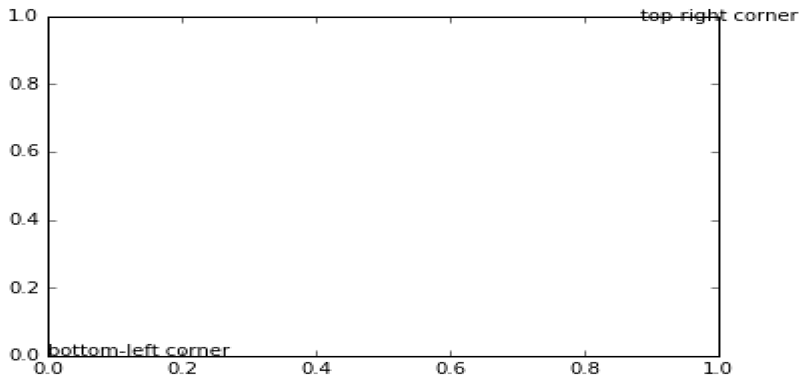
text 함수 : 위치 지정 1

text함수에 위치지정 파라미터 수직방향(va: top, bottom, center, baseline), 수평방향(ha :center, right, left')로 표시



```
import matplotlib.pyplot as plt
txt = plt.text(0, 0, 'bottom-left corner')
txt1 = plt.text(1, 1, 'top-right corner',ha='center', va='center')
print txt
print txt1
plt.show()
```

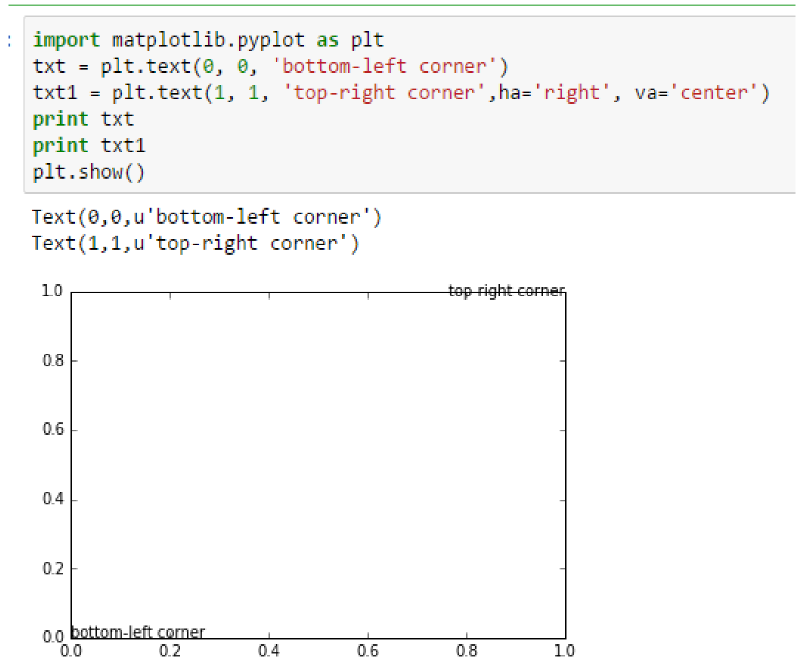
```
Text(0,0,u'bottom-left corner')
Text(1,1,u'top-right corner')
```



text 함수 : 위치 지정 2

text함수에 수평방향은 위치를 표시할 경우 우리가 보는 반대방향에 표시 됨

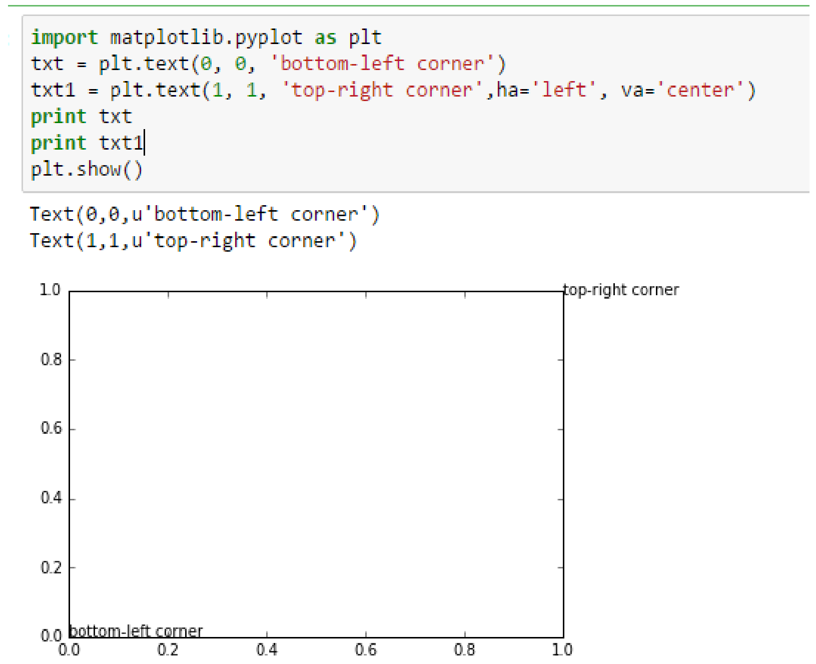
오른쪽



```
import matplotlib.pyplot as plt
txt = plt.text(0, 0, 'bottom-left corner')
txt1 = plt.text(1, 1, 'top-right corner',ha='right', va='center')
print txt
print txt1
plt.show()
```

```
Text(0,0,u'bottom-left corner')
Text(1,1,u'top-right corner')
```

왼쪽



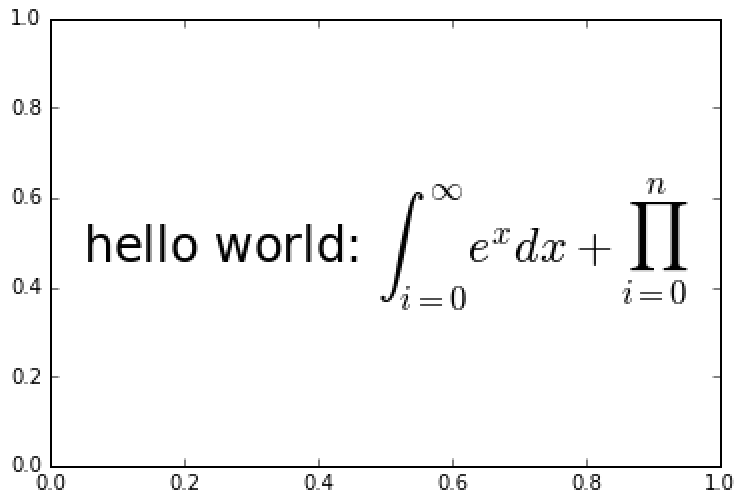
```
import matplotlib.pyplot as plt
txt = plt.text(0, 0, 'bottom-left corner')
txt1 = plt.text(1, 1, 'top-right corner',ha='left', va='center')
print txt
print txt1
plt.show()
```

```
Text(0,0,u'bottom-left corner')
Text(1,1,u'top-right corner')
```

text 함수 :latex로 기호 표시

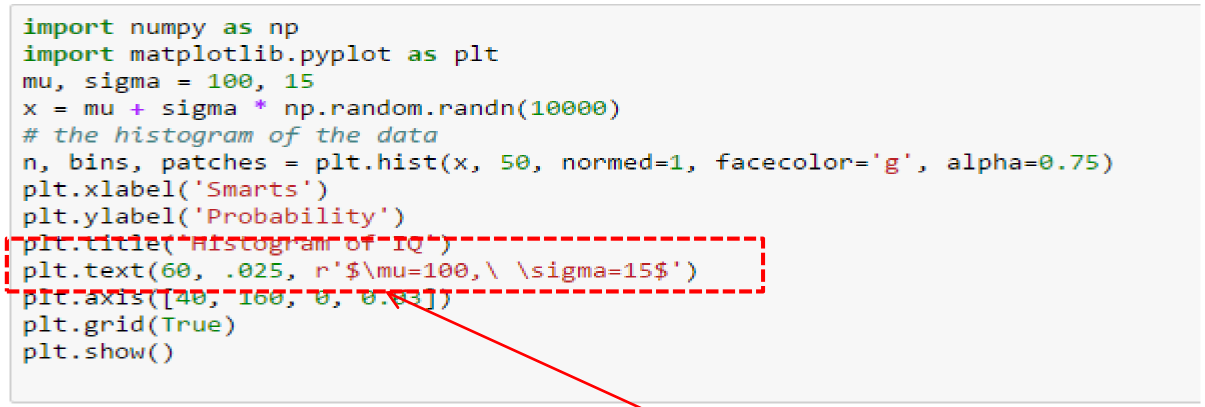
문자열 내의 기호는 latex 방식에 위해 표시

```
import numpy as np
import matplotlib.pyplot as plt
plt.text(0.5, 0.5, 'hello world: $\int_{i=0}^\infty e^x dx + \prod_{i=0}^n$', size=24, ha='center', va='center')
plt.show()
```

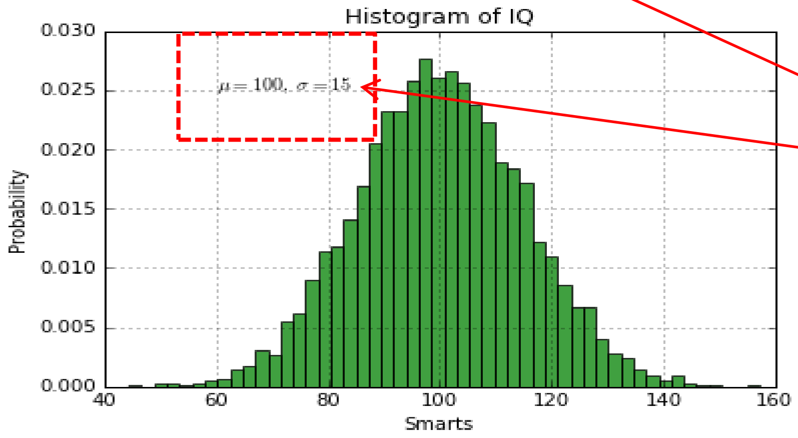


text 함수 : text 붙이기

그래프 내에 text를 사용해서 입력하기



```
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()
```

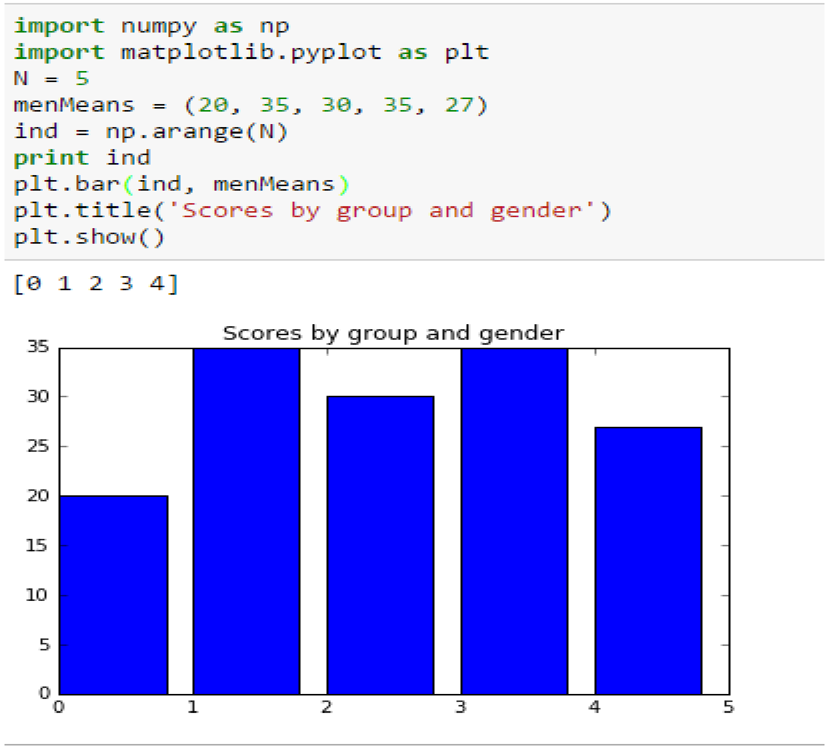


텍스트에 대해 입력

title

title 함수 : 제목 붙이기

그래프에 제목을 표시

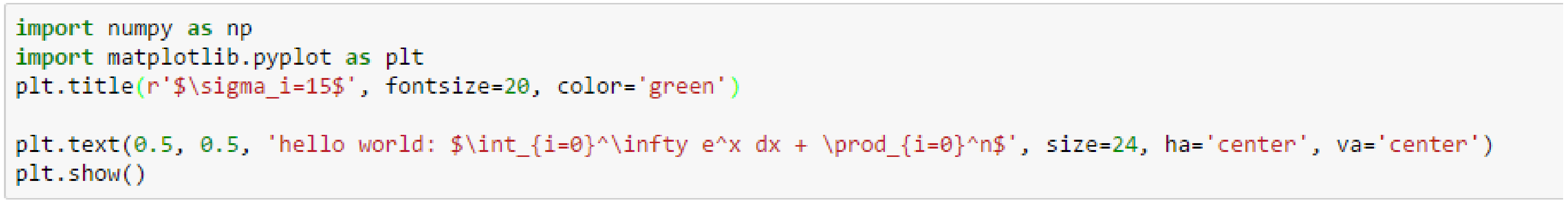


```
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans)
plt.title('Scores by group and gender')
plt.show()
```

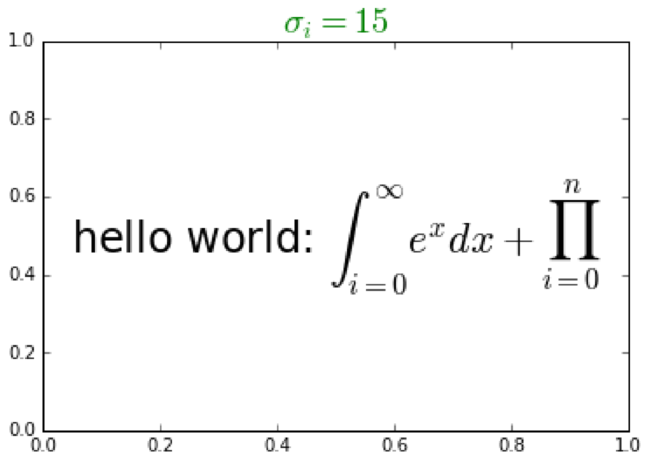
```
[0 1 2 3 4]
```

title 함수 : font/color 처리

Latex로 정의한 문자열에 대해 fontsize와 color 처리



```
import numpy as np
import matplotlib.pyplot as plt
plt.title(r'$\sigma_i=15$', fontsize=20, color='green')

plt.text(0.5, 0.5, 'hello world: $\int_{i=0}^\infty e^x dx + \prod_{i=0}^n$', size=24, ha='center', va='center')
plt.show()
```



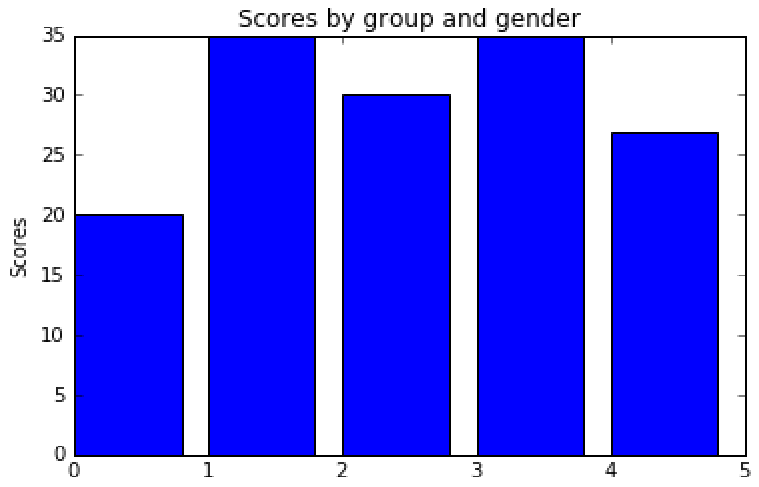
label

ylabel 함수 : label 붙이기

y축 그래프에 의미를 부여하기

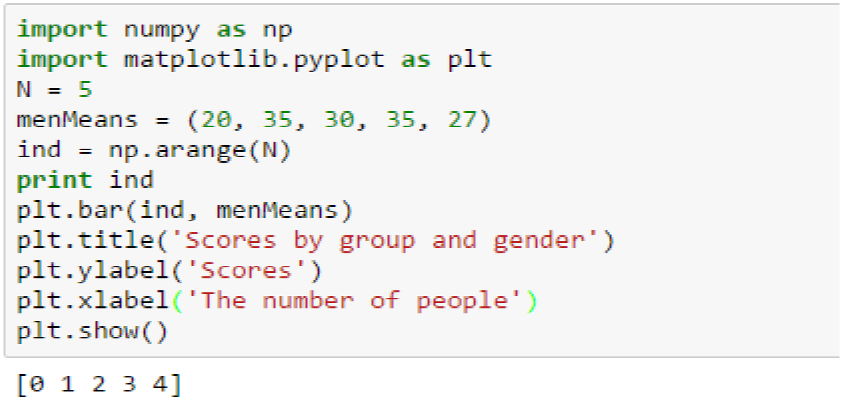
```
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans)
plt.title('Scores by group and gender')
plt.ylabel('Scores')
plt.show()
```

```
[0 1 2 3 4]
```



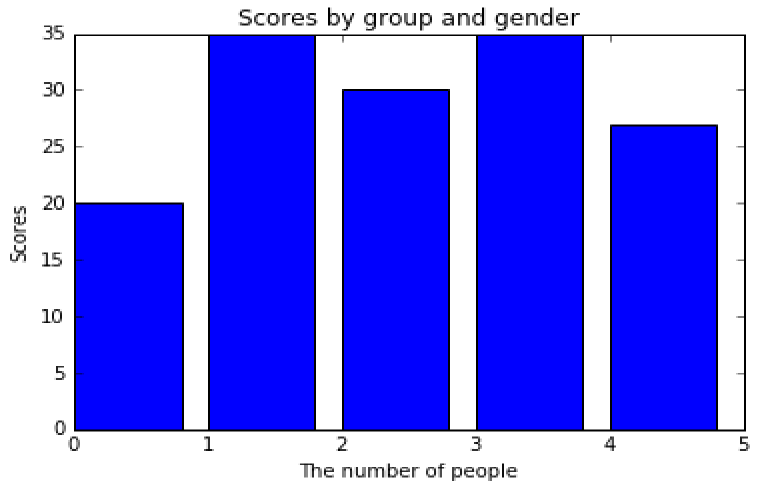
xlabel 함수 : label 붙이기

x축 그래프에 의미적인 레이블을 부여하기



```
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans)
plt.title('Scores by group and gender')
plt.ylabel('Scores')
plt.xlabel('The number of people')
plt.show()
```

```
[0 1 2 3 4]
```

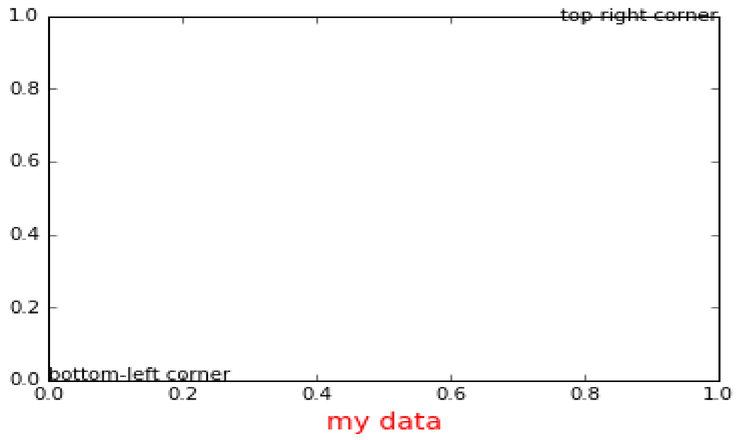


xlabel 함수 : font/color

x축 그래프에 label에 fontsize와 font color 변경하기

```
import matplotlib.pyplot as plt
txt = plt.text(0, 0, 'bottom-left corner')
txt1 = plt.text(1, 1, 'top-right corner',ha='right', va='center')
print txt
print txt1
t = plt.xlabel('my data', fontsize=14, color='red')

plt.show()
```

```
Text(0,0,u'bottom-left corner')
Text(1,1,u'top-right corner')
```

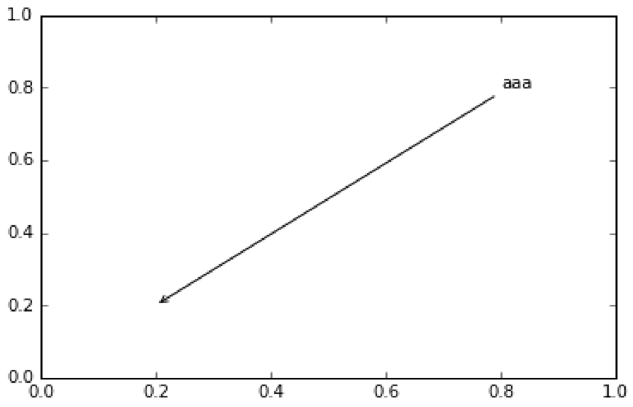


annotate

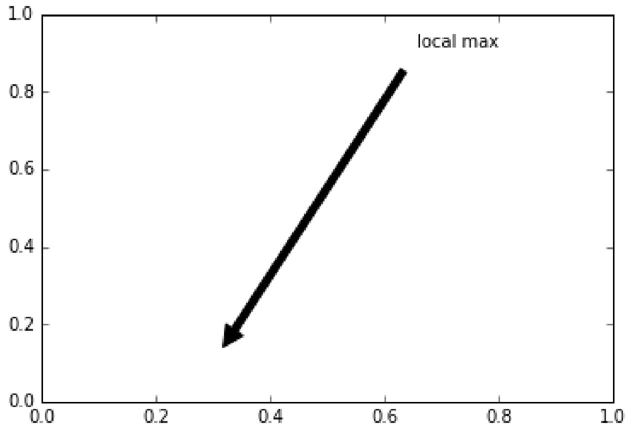
annotate 함수 : 기초

annotate 함수는 문장열, xy(화살표 끝 지시), xytext(문자열 시작 위치), arrowpros(화살표) 그래프에 주석을 표시

```
import matplotlib.pyplot as plt
plt.annotate("aaa",
            xy=(0.2, 0.2), xycoords='data',
            xytext=(0.8, 0.8), textcoords='data',
            arrowprops=dict(arrowstyle="->",
                            connectionstyle="arc3"),
            )

plt.show()
```

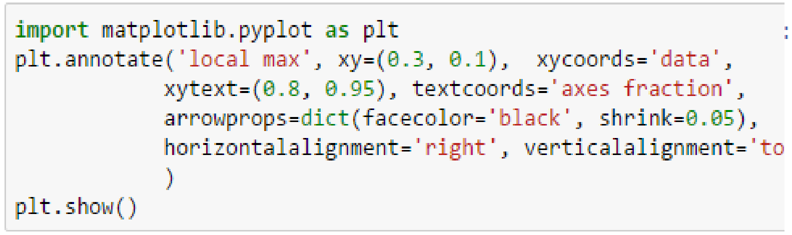


```
import matplotlib.pyplot as plt
plt.annotate('local max', xy=(0.3, 0.1),  xycoords='data',
            xytext=(0.8, 0.95), textcoords='axes fraction',
            arrowprops=dict(facecolor='black', shrink=0.05),
            horizontalalignment='right', verticalalignment='top',
            )
plt.show()
```

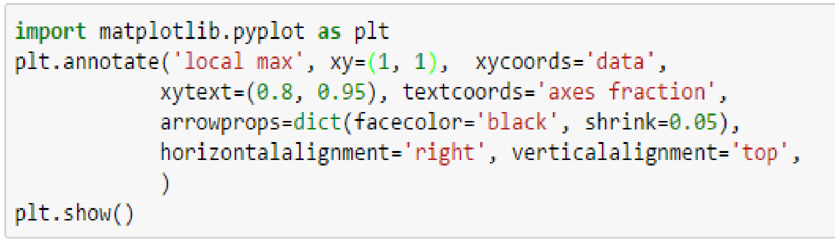


xycoords/textcoords :1

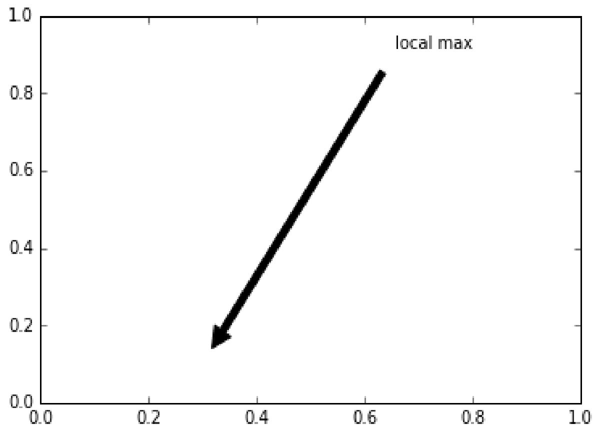
"axes fraction"으로 지정시 xy 좌표가 1보다 작으면 아래 방향으로 1보다 크거나 같으면 위로 가르킴

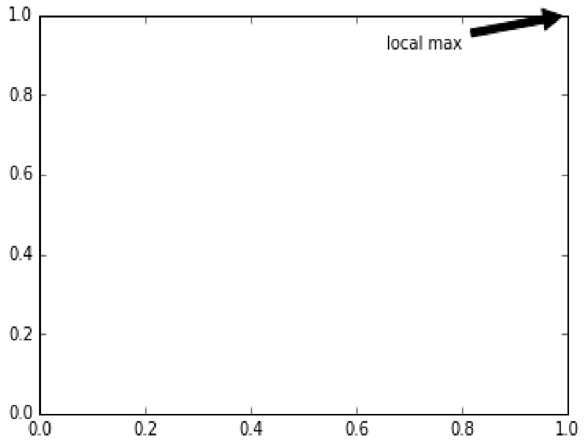


```
import matplotlib.pyplot as plt
plt.annotate('local max', xy=(0.3, 0.1),  xycoords='data',
            xytext=(0.8, 0.95), textcoords='axes fraction',
            arrowprops=dict(facecolor='black', shrink=0.05),
            horizontalalignment='right', verticalalignment='to
            )
plt.show()
```



```
import matplotlib.pyplot as plt
plt.annotate('local max', xy=(1, 1),  xycoords='data',
            xytext=(0.8, 0.95), textcoords='axes fraction',
            arrowprops=dict(facecolor='black', shrink=0.05),
            horizontalalignment='right', verticalalignment='top',
            )
plt.show()
```





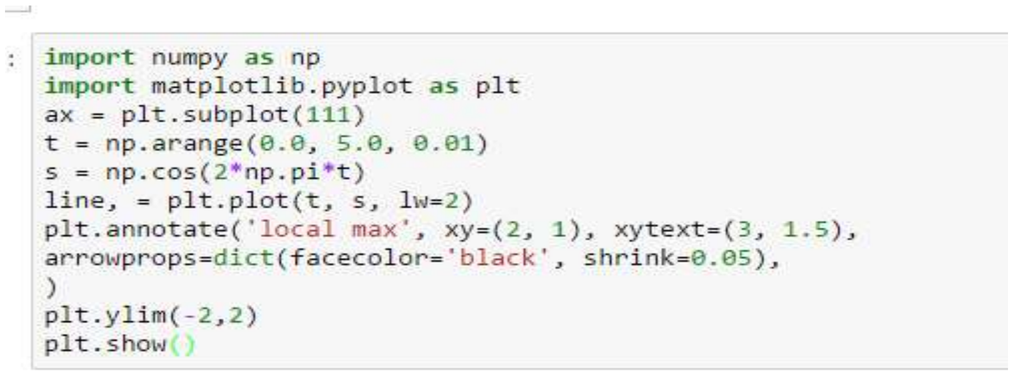
xycoords/textcoords : 값 설명

xycoords/textcoords 내의 값에 대한 설명

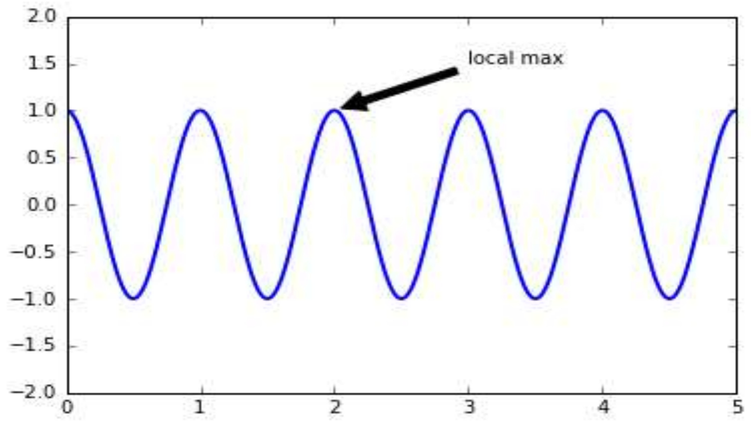
argument	coordinate system
‘figure points’	points from the lower left corner of the figure
‘figure pixels’	pixels from the lower left corner of the figure
‘figure fraction’	0,0 is lower left of figure and 1,1 is upper right
‘axes points’	points from lower left corner of axes
‘axes pixels’	pixels from lower left corner of axes
‘axes fraction’	0,0 is lower left of axes and 1,1 is upper right
‘data’	use the axes data coordinate system

annotate 함수 : 실행

annotate 함수는 그래프에 주석을 표시



```
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()
```



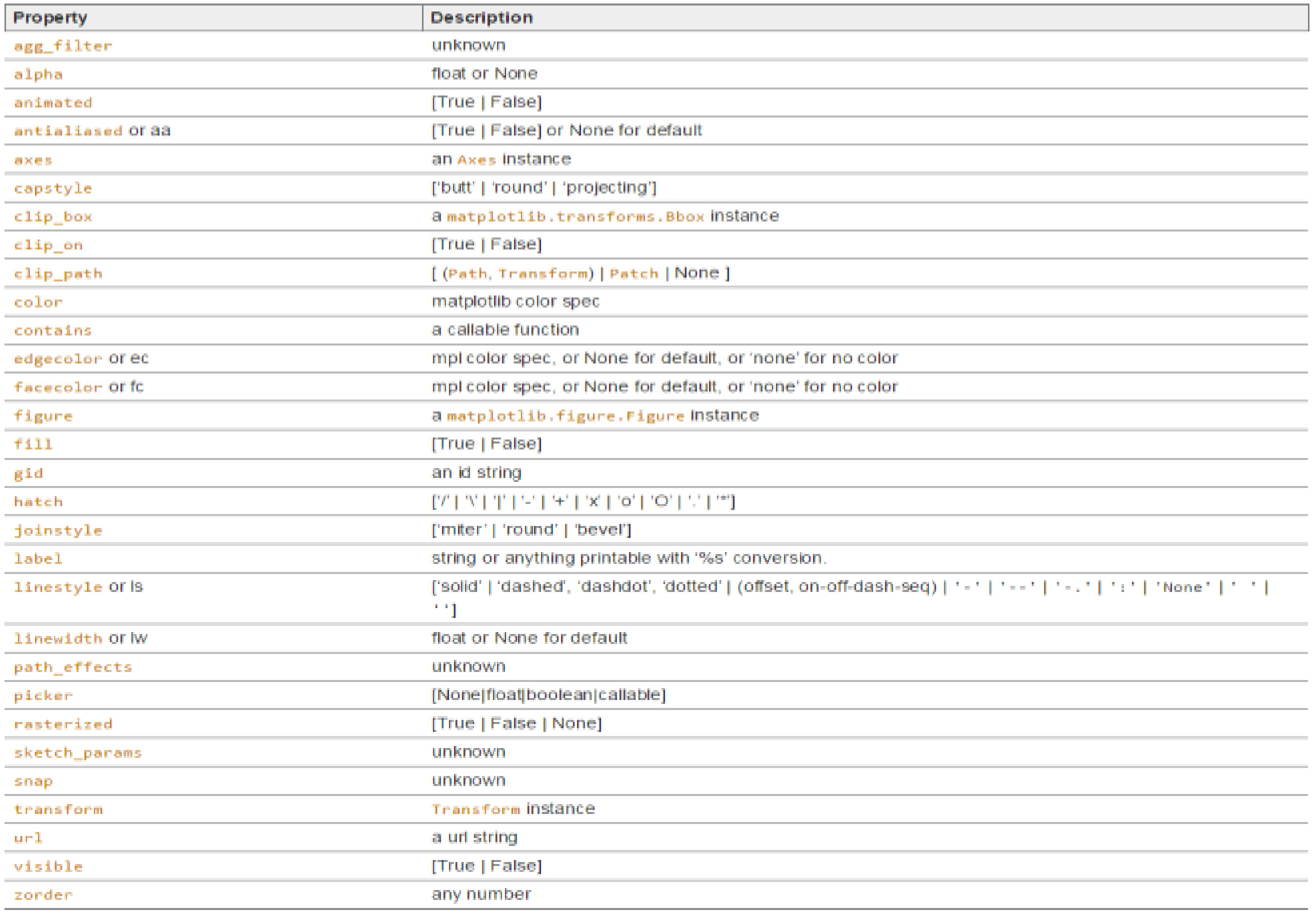
annotate 함수 : arrowpros

arrowpros(화살표) 이 주요 파라미터 및 Polygon 파라미터 이용

arrowprops key	description
width	the width of the arrow in points
frac	the fraction of the arrow length occupied by the head
headwidth	the width of the base of the arrow head in points
shrink	move the tip and base some percent away from the annotated poi nt and text
**kwargs	any key for matplotlib.patches.Polygon, e.g., facecolor

matplotlib.patches.Polygon,

도형을 그리는 클래스의 속성들

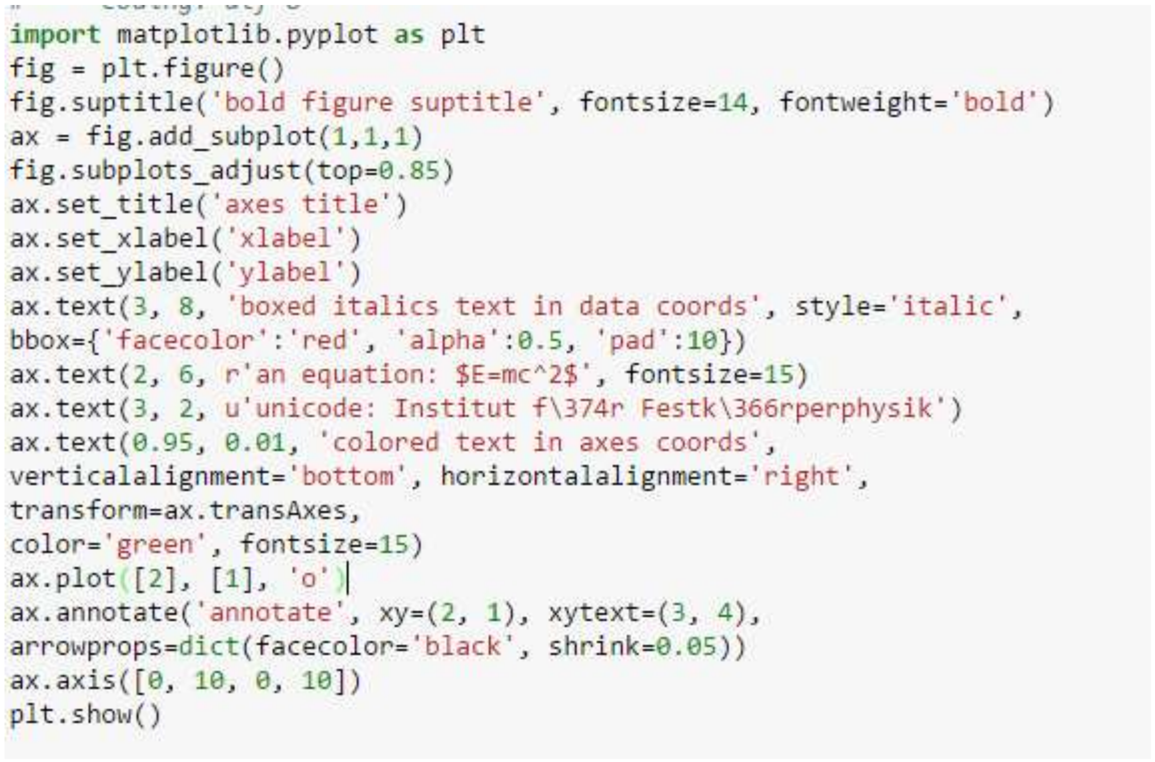


Property	Description
agg_filter	unknown
alpha	float or None
animated	[True | False]
antialiased or aa	[True | False] or None for default
axes	an Axes instance
capstyle	['butt' | 'round' | 'projecting']
clip_box	a matplotlib.transforms.Bbox instance
clip_on	[True | False]
clip_path	[(Path, Transform) | Patch | None]
color	matplotlib color spec
contains	a callable function
edgecolor or ec	mpl color spec, or None for default, or 'none' for no color
facecolor or fc	mpl color spec, or None for default, or 'none' for no color
figure	a matplotlib.figure.Figure instance
fill	[True | False]
gid	an id string
hatch	['/' | '\\' | '|' | '-' | '+' | 'x' | 'o' | 'O' | '.' | '*']
joinstyle	['miter' | 'round' | 'bevel']
label	string or anything printable with '%s' conversion.
linestyle or ls	['solid' | 'dashed', 'dashdot', 'dotted' | (offset, on-off-dash-seq) | '-' | '--' | '-.' | ':' | 'None' | ' ' | '']
linewidth or lw	float or None for default
path_effects	unknown
picker	[None|float|boolean|callable]
rasterized	[True | False | None]
sketch_params	unknown
snap	unknown
transform	Transform instance
url	a url string
visible	[True | False]
zorder	any number

Method 사용하기

Figure 메소드 사용

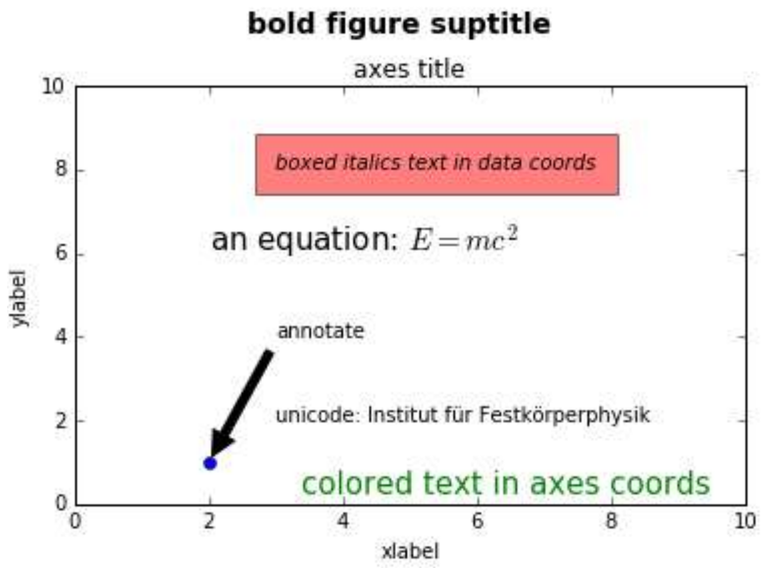
Figure 내부의 메소드를 해서 사용하기



```
import matplotlib.pyplot as plt
fig = plt.figure()
fig.suptitle('bold figure suptitle', fontsize=14, fontweight='bold')
ax = fig.add_subplot(1,1,1)
fig.subplots_adjust(top=0.85)
ax.set_title('axes title')
ax.set_xlabel('xlabel')
ax.set_ylabel('ylabel')
ax.text(3, 8, 'boxed italics text in data coords', style='italic',
bbox={'facecolor':'red', 'alpha':0.5, 'pad':10})
ax.text(2, 6, r'an equation: $E=mc^2$', fontsize=15)
ax.text(3, 2, u'unicode: Institut f\374r Festk\366rperphysik')
ax.text(0.95, 0.01, 'colored text in axes coords',
verticalalignment='bottom', horizontalalignment='right',
transform=ax.transAxes,
color='green', fontsize=15)
ax.plot([2], [1], 'o')
ax.annotate('annotate', xy=(2, 1), xytext=(3, 4),
arrowprops=dict(facecolor='black', shrink=0.05))
ax.axis([0, 10, 0, 10])
plt.show()
```

Figure 메소드 사용: 결과

결과



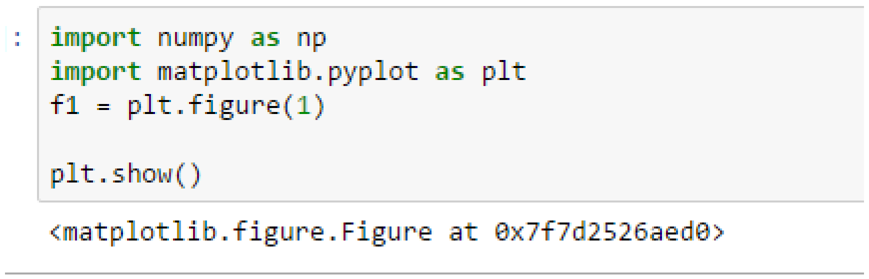
MATPLOTLIB 화면 꾸미기

Moon Yong Joon

figure/subplot

figure 함수 : 기초

하나의 화면에 그래프를 여러 개 그리기위해서 는 figure 함수를 지정해서 Figure 객체를 생성

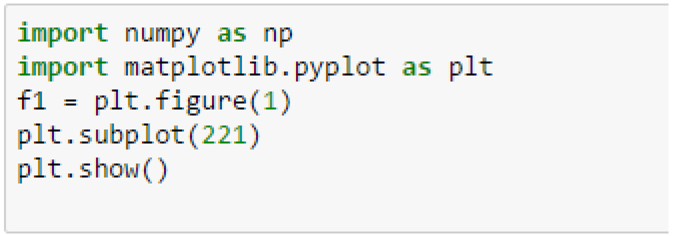


```
import numpy as np
import matplotlib.pyplot as plt
f1 = plt.figure(1)

plt.show()
```

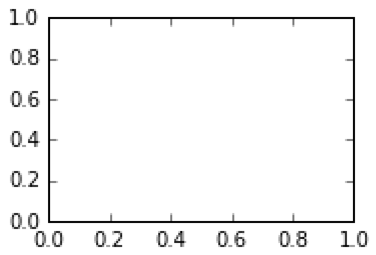
```
<matplotlib.figure.Figure at 0x7f7d2526aed0>
```

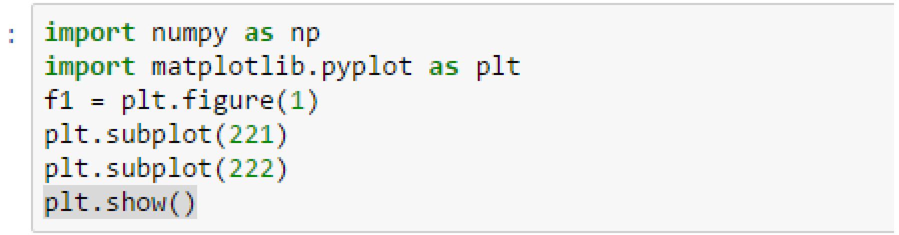
subplot함수 : 캔버스를 분리 1

하나의 화면에 그래프를 여러 개 그리기

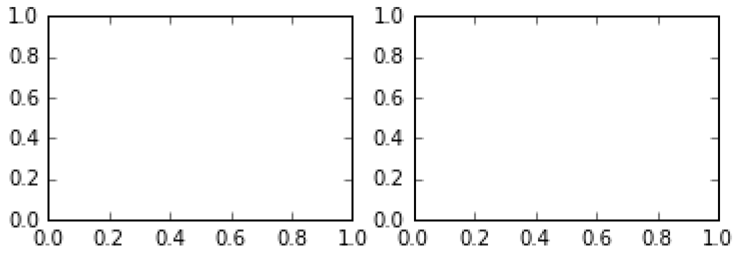


```python
import numpy as np
import matplotlib.pyplot as plt
f1 = plt.figure(1)
plt.subplot(221)
plt.show()
```



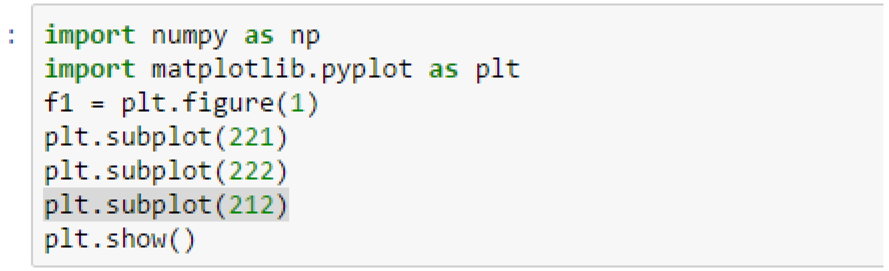


```python
import numpy as np
import matplotlib.pyplot as plt
f1 = plt.figure(1)
plt.subplot(221)
plt.subplot(222)
plt.show()
```

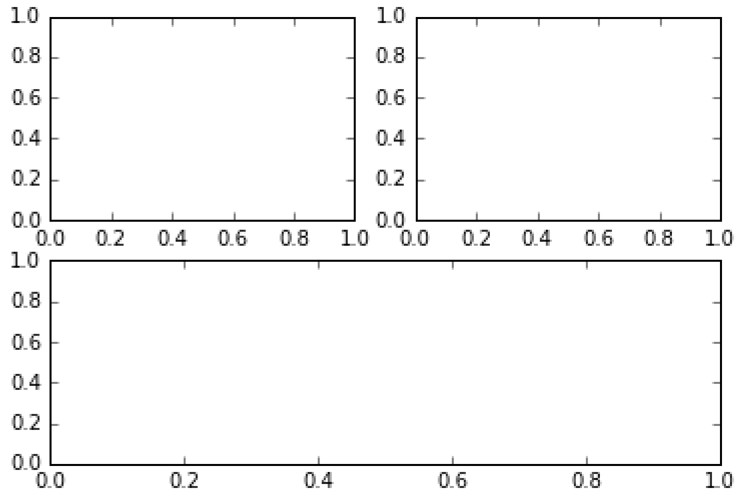


subplot함수 : 캔버스를 분리 2

하나의 화면에 그래프를 여러 개 그리기

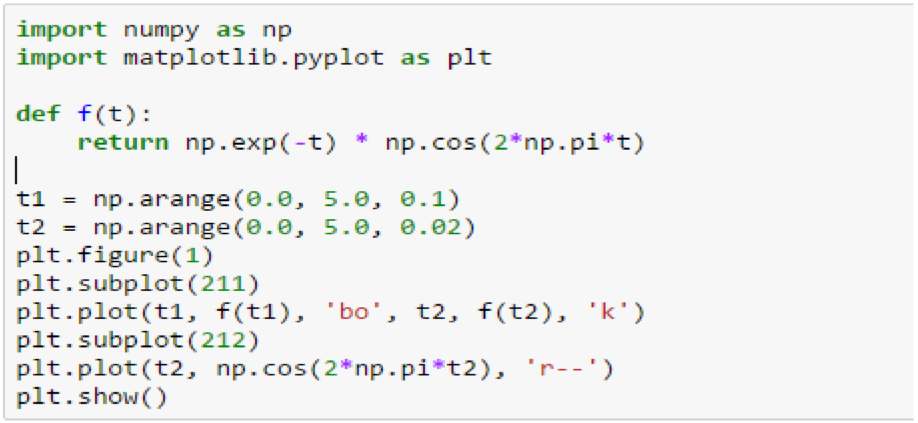


```
import numpy as np
import matplotlib.pyplot as plt
f1 = plt.figure(1)
plt.subplot(221)
plt.subplot(222)
plt.subplot(212)
plt.show()
```

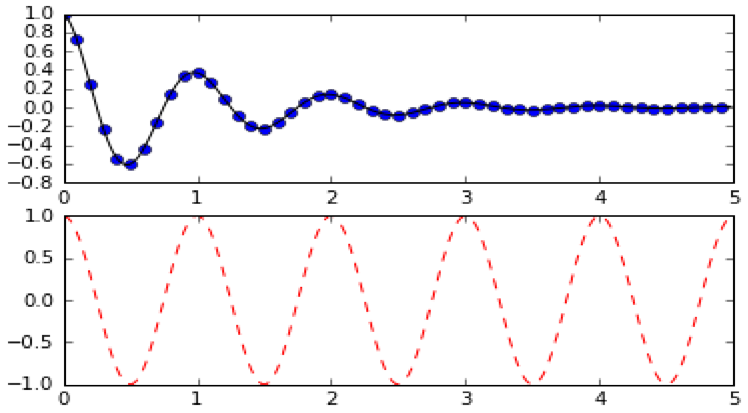


figure/subplot함수 : 예시

하나의 화면에 그래프를 여러 개 그리기

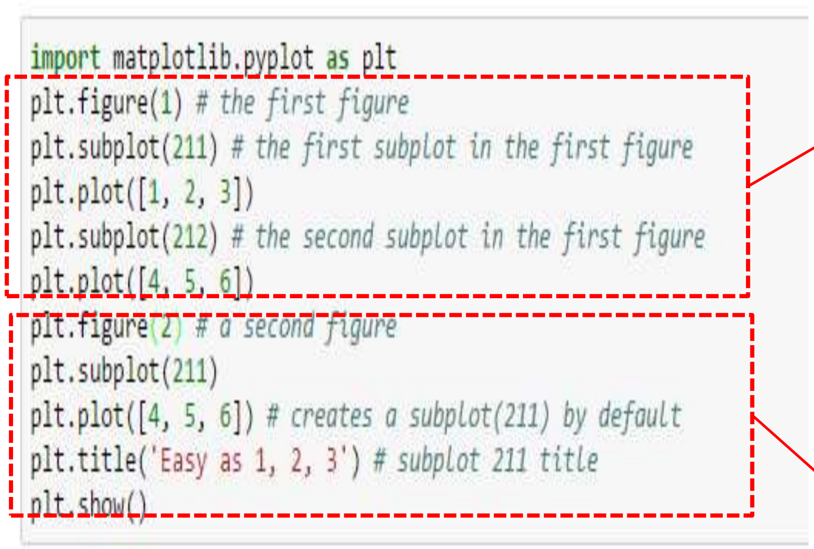


```python
import numpy as np
import matplotlib.pyplot as plt

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)
plt.subplot(211)
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()
```

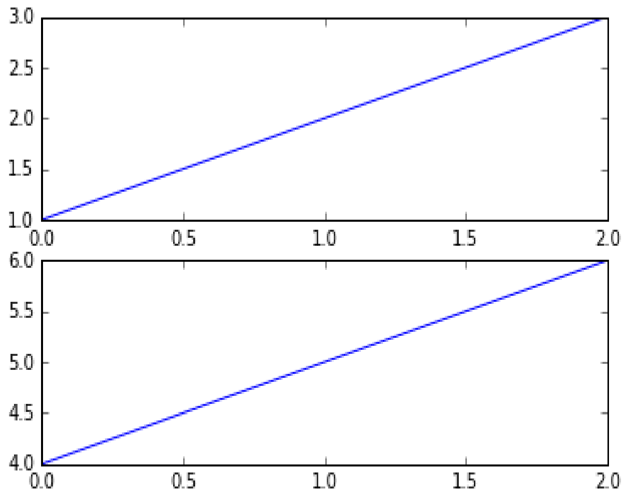


figure/subplot: 여러 개 분리

두개의 캔버스로 분리해서 subplot으로 그래프 그리기



```
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.subplot(211)
plt.plot([4, 5, 6]) # creates a subplot(211) by default
plt.title('Easy as 1, 2, 3') # subplot 211 title
plt.show()
```



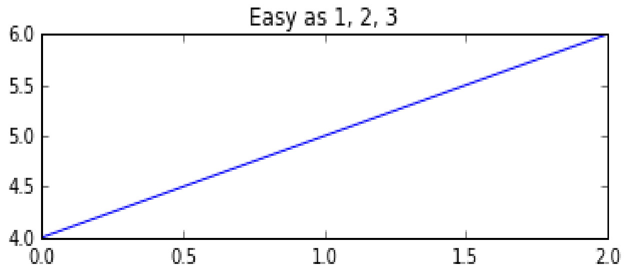
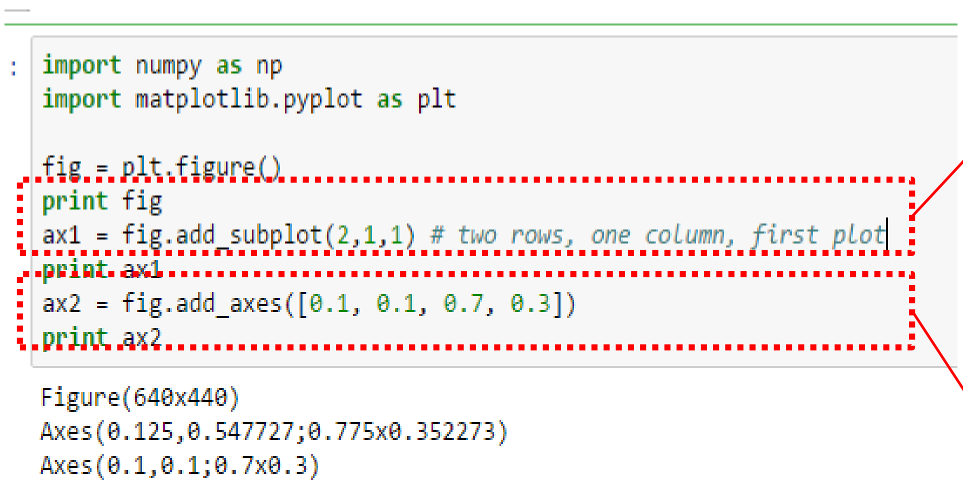


Figure class

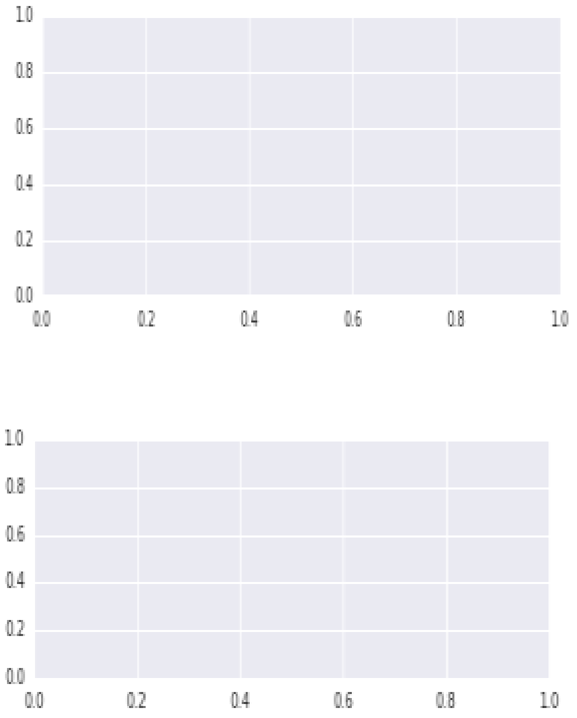
add_subplot/add_axes 메소드

Figure 하나를 생성하고 2개의 내부 Axes 객체 생성



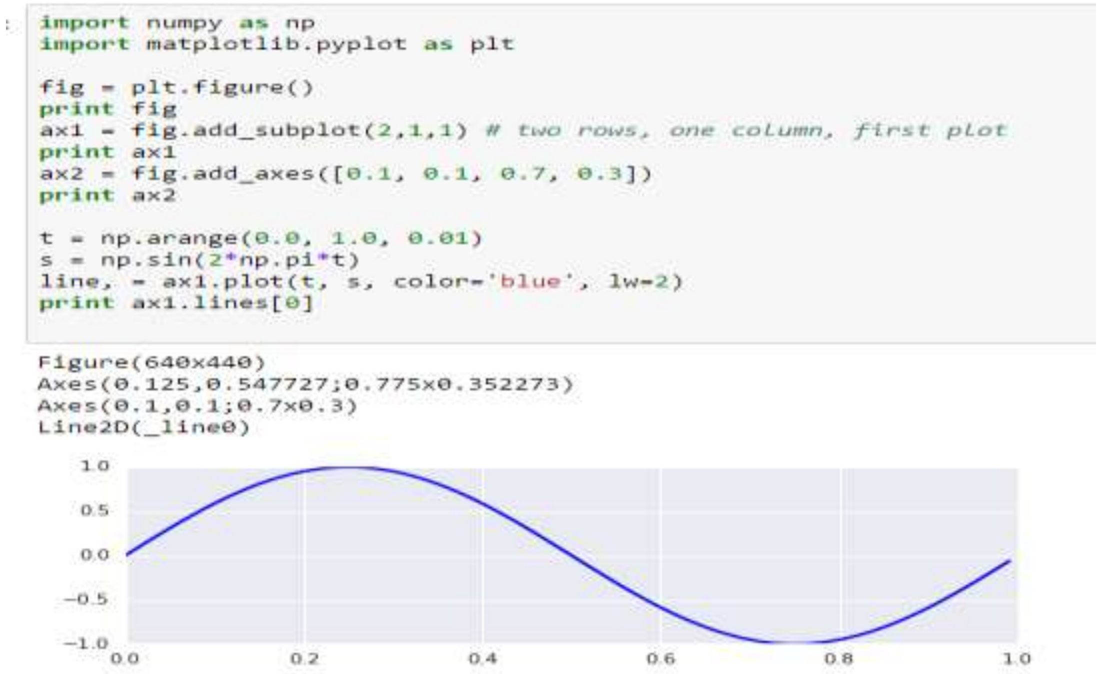
```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()
print fig
ax1 = fig.add_subplot(2,1,1) # two rows, one column, first plot
print ax1
ax2 = fig.add_axes([0.1, 0.1, 0.7, 0.3])
print ax2
```

```
Figure(640x440)
Axes(0.125,0.547727;0.775x0.352273)
Axes(0.1,0.1;0.7x0.3)
```



첫번째 그래프 표시

Axes로 생성된 ax1에 plot 할당.
ax1.lines[0] 내의 저장된 것을 조회

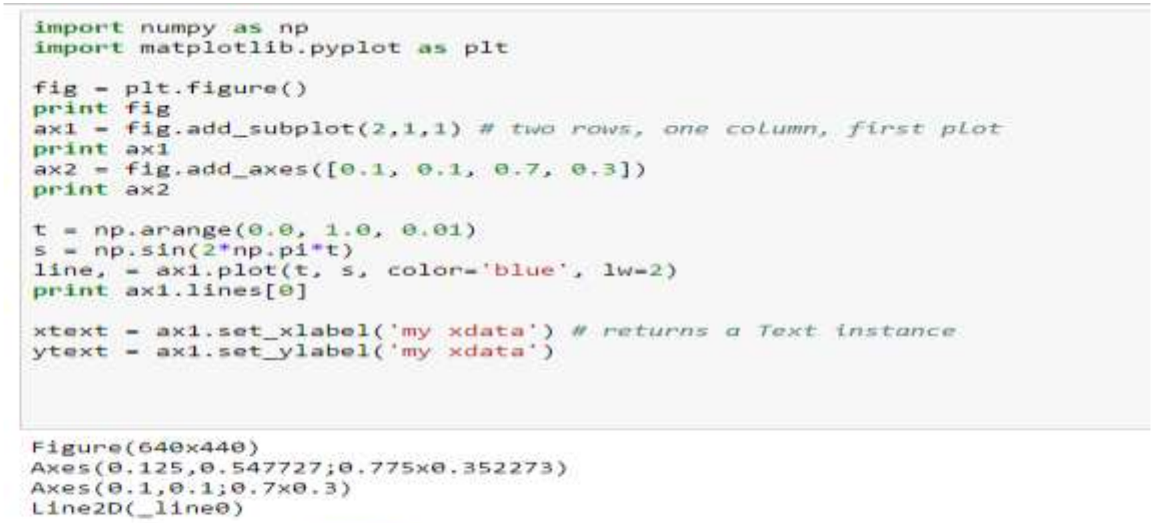


```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()
print fig
ax1 = fig.add_subplot(2,1,1) # two rows, one column, first plot
print ax1
ax2 = fig.add_axes([0.1, 0.1, 0.7, 0.3])
print ax2

t = np.arange(0.0, 1.0, 0.01)
s = np.sin(2*np.pi*t)
line, = ax1.plot(t, s, color='blue', lw=2)
print ax1.lines[0]
```

```
Figure(640x440)
Axes(0.125,0.547727;0.775x0.352273)
Axes(0.1,0.1;0.7x0.3)
Line2D(_line0)
```

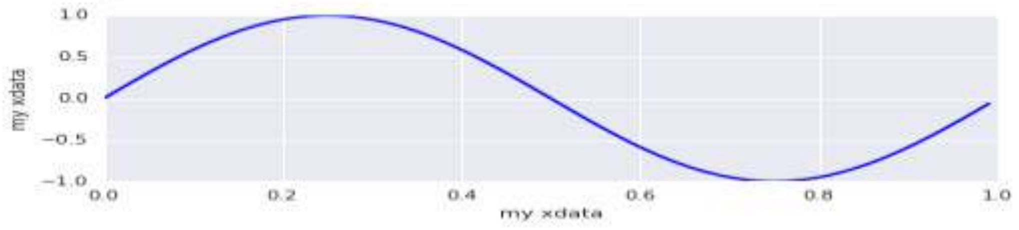
첫번째 그래프에 label 추가

Axes로 생성된 ax1에 set_xlabel, set_ylabel 로 레이블 추가



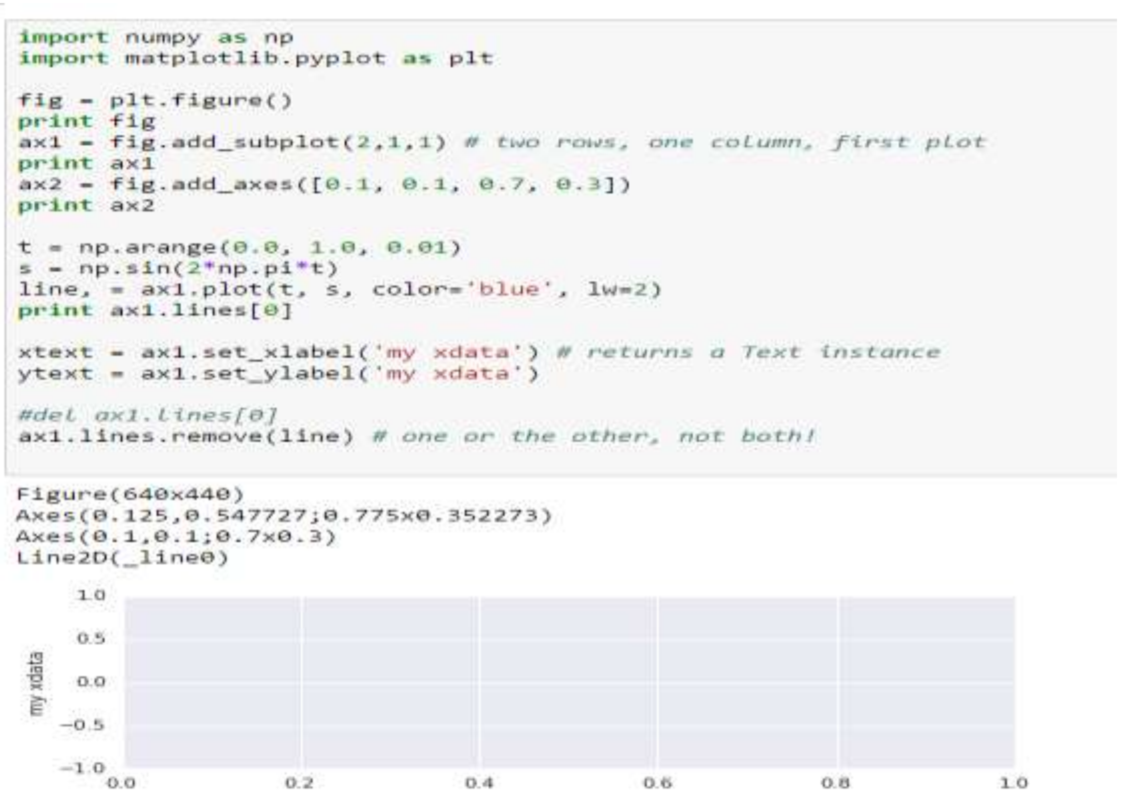
```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()
print fig
ax1 = fig.add_subplot(2,1,1) # two rows, one column, first plot
print ax1
ax2 = fig.add_axes([0.1, 0.1, 0.7, 0.3])
print ax2

t = np.arange(0.0, 1.0, 0.01)
s = np.sin(2*np.pi*t)
line, = ax1.plot(t, s, color='blue', lw=2)
print ax1.lines[0]

xtext = ax1.set_xlabel('my xdata') # returns a Text instance
ytext = ax1.set_ylabel('my xdata')
```

```
Figure(640x440)
Axes(0.125,0.547727;0.775x0.352273)
Axes(0.1,0.1;0.7x0.3)
Line2D(_line0)
```



첫번째 그래프 지우려면

del ax1.lines[0], ax1.lines.remove(line)으로 그래프 삭제



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

fig = plt.figure()
print fig
ax1 = fig.add_subplot(2,1,1) # two rows, one column, first plot
print ax1
ax2 = fig.add_axes([0.1, 0.1, 0.7, 0.3])
print ax2

t = np.arange(0.0, 1.0, 0.01)
s = np.sin(2*np.pi*t)
line, = ax1.plot(t, s, color='blue', lw=2)
print ax1.lines[0]

xtext = ax1.set_xlabel('my xdata') # returns a Text instance
ytext = ax1.set_ylabel('my xdata')

#del ax1.lines[0]
ax1.lines.remove(line) # one or the other, not both!
```

```
Figure(640x440)
Axes(0.125,0.547727;0.775x0.352273)
Axes(0.1,0.1;0.7x0.3)
Line2D(_line0)
```

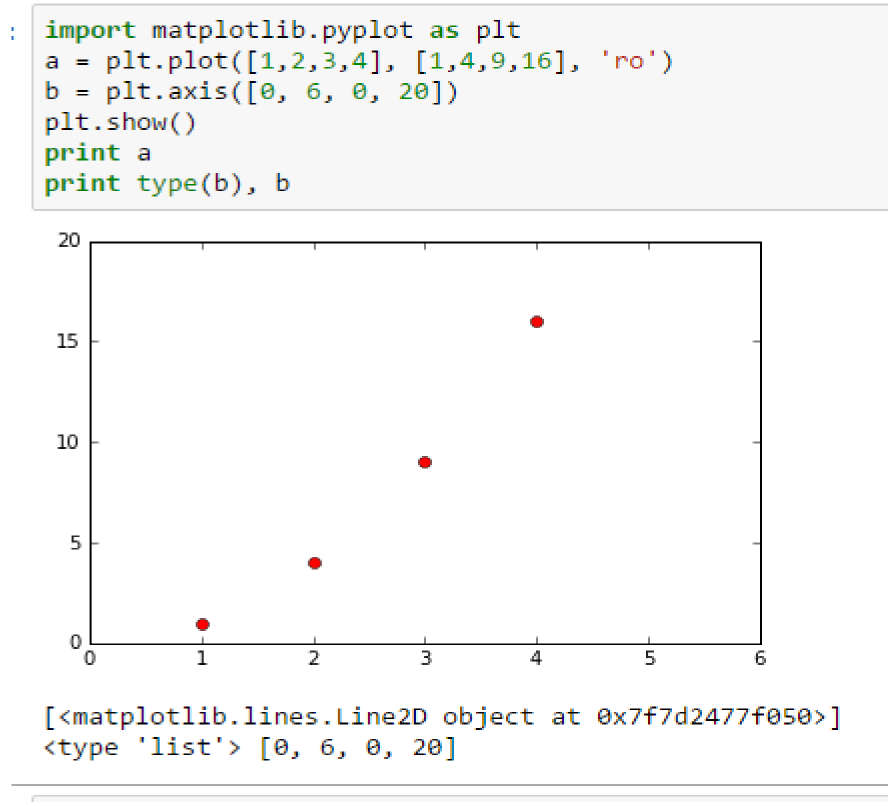
MATPLOTLIB 좌표 그리기

Moon Yong Joon

axis

axis 함수 이해하기

axis 함수는 리스트의 값을 그대로 표시하고 앞의 2자리는 x축, 뒤에 2자리는 y축을 표시



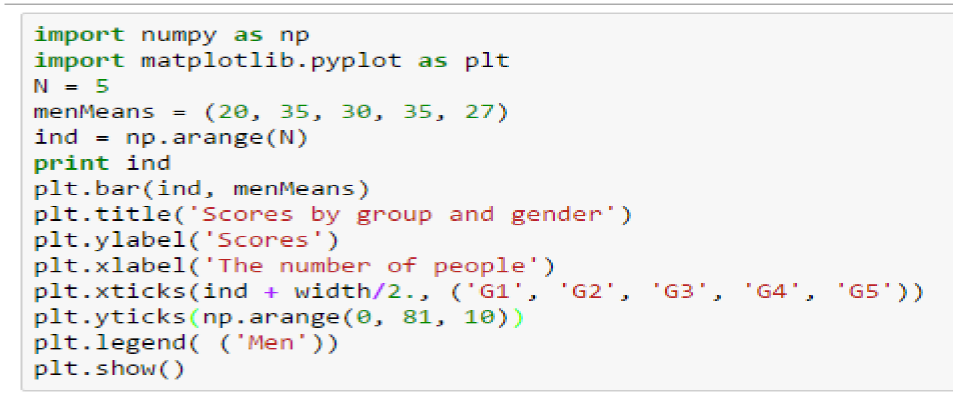
```
import matplotlib.pyplot as plt
a = plt.plot([1,2,3,4], [1,4,9,16], 'ro')
b = plt.axis([0, 6, 0, 20])
plt.show()
print a
print type(b), b
```

```
[<matplotlib.lines.Line2D object at 0x7f7d2477f050>]
<type 'list'> [0, 6, 0, 20]
```

ticks

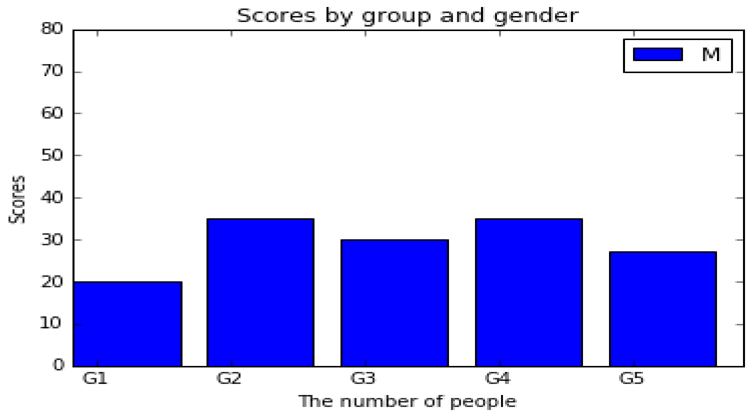
ticks 함수 : 축 넣기

xticks, yticks 함수를 이용해서 세부 값을 부여



```
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans)
plt.title('Scores by group and gender')
plt.ylabel('Scores')
plt.xlabel('The number of people')
plt.xticks(ind + width/2., ('G1', 'G2', 'G3', 'G4', 'G5'))
plt.yticks(np.arange(0, 81, 10))
plt.legend( ('Men'))
plt.show()
```

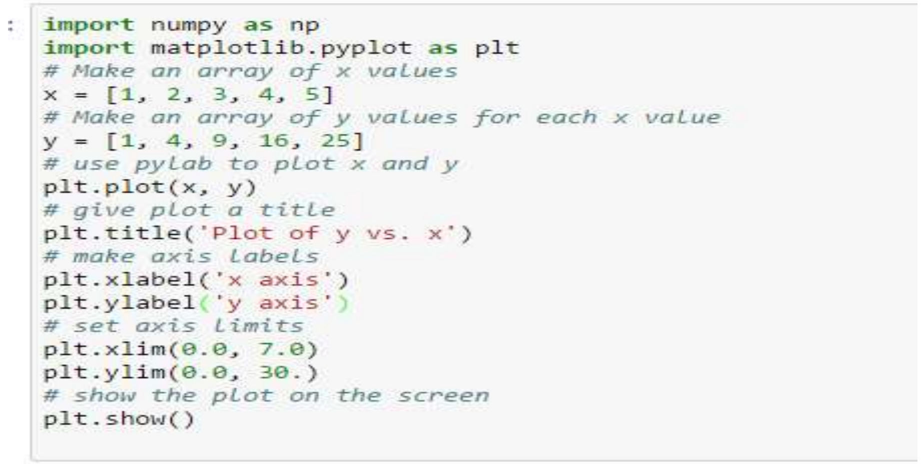
```
[0 1 2 3 4]
```



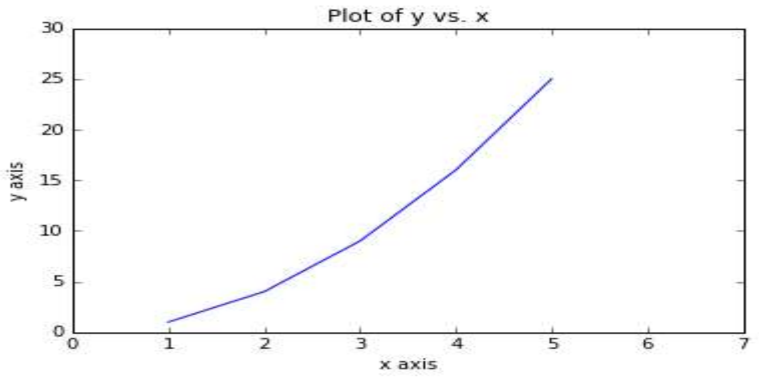
limit

lim 함수 : 축 넣기

xlim, ylim 함수를 이용해서 축내의 범위 값을 부여



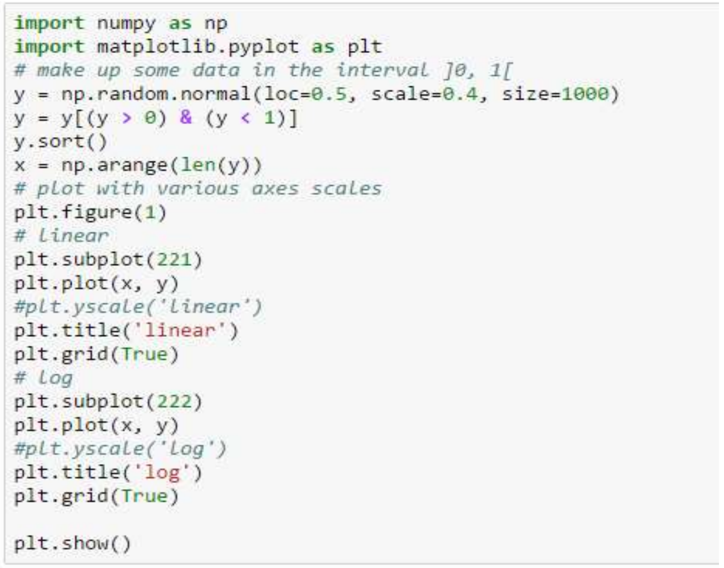
```
import numpy as np
import matplotlib.pyplot as plt
# Make an array of x values
x = [1, 2, 3, 4, 5]
# Make an array of y values for each x value
y = [1, 4, 9, 16, 25]
# use pylab to plot x and y
plt.plot(x, y)
# give plot a title
plt.title('Plot of y vs. x')
# make axis labels
plt.xlabel('x axis')
plt.ylabel('y axis')
# set axis limits
plt.xlim(0.0, 7.0)
plt.ylim(0.0, 30.)
# show the plot on the screen
plt.show()
```



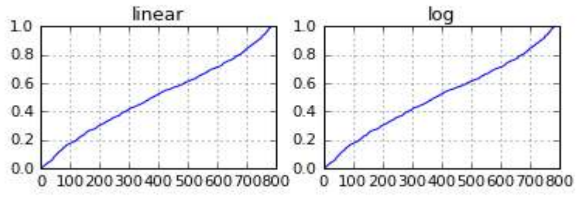
scale

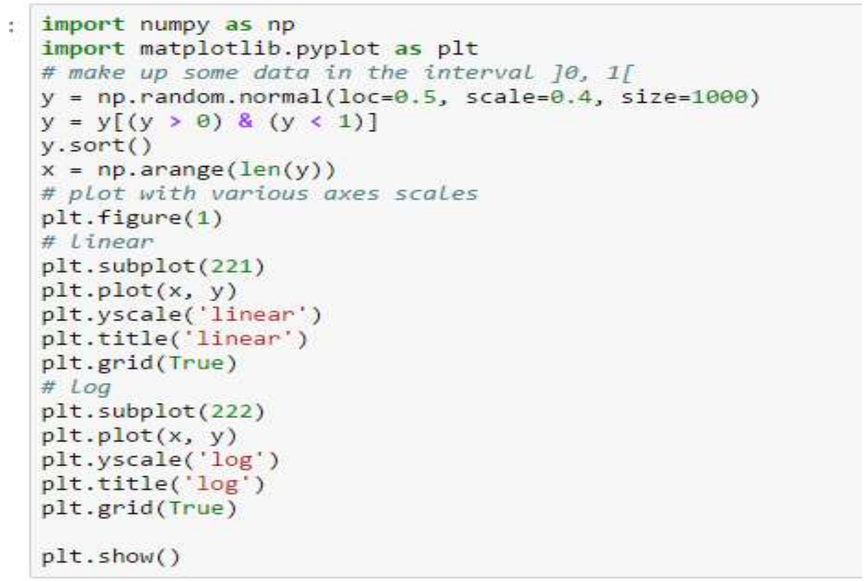
scale 함수 : 축 자동 변환

yscale을 막고 처리하면 고정축을 가지고 표시 하지만 yscale 처리하면 y축에 스케일처리 됨

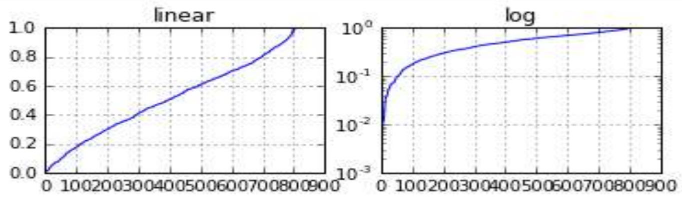


```
import numpy as np
import matplotlib.pyplot as plt
# make up some data in the interval ]0, 1[
y = np.random.normal(loc=0.5, scale=0.4, size=1000)
y = y[(y > 0) & (y < 1)]
y.sort()
x = np.arange(len(y))
# plot with various axes scales
plt.figure(1)
# linear
plt.subplot(221)
plt.plot(x, y)
#plt.yscale('linear')
plt.title('linear')
plt.grid(True)
# log
plt.subplot(222)
plt.plot(x, y)
#plt.yscale('log')
plt.title('log')
plt.grid(True)

plt.show()
```





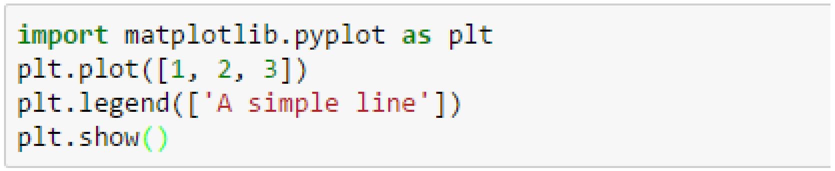
```
import numpy as np
import matplotlib.pyplot as plt
# make up some data in the interval ]0, 1[
y = np.random.normal(loc=0.5, scale=0.4, size=1000)
y = y[(y > 0) & (y < 1)]
y.sort()
x = np.arange(len(y))
# plot with various axes scales
plt.figure(1)
# linear
plt.subplot(221)
plt.plot(x, y)
plt.yscale('linear')
plt.title('linear')
plt.grid(True)
# log
plt.subplot(222)
plt.plot(x, y)
plt.yscale('log')
plt.title('log')
plt.grid(True)

plt.show()
```



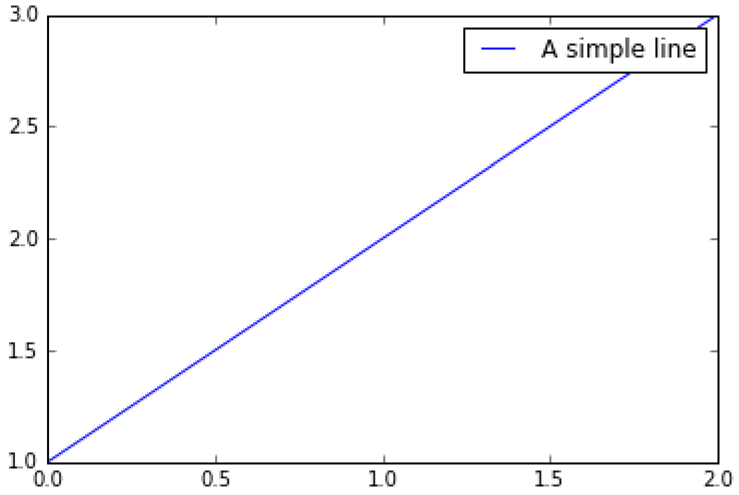
legend

기본

그래프에 범주를 표시



```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3])
plt.legend(['A simple line'])
plt.show()
```

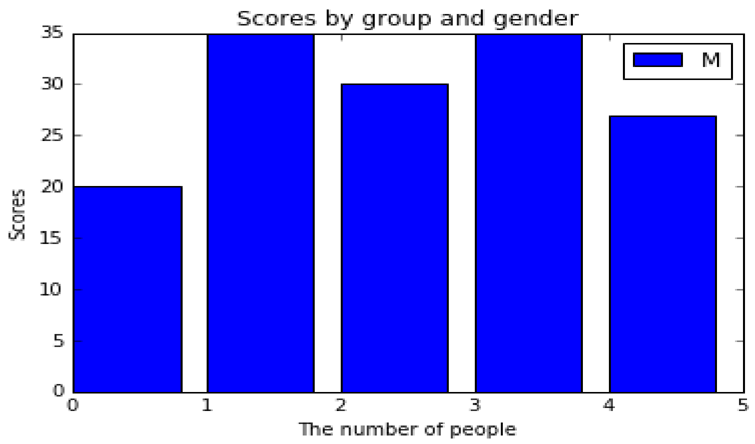


범주 붙이기

그래프에 범주를 표시

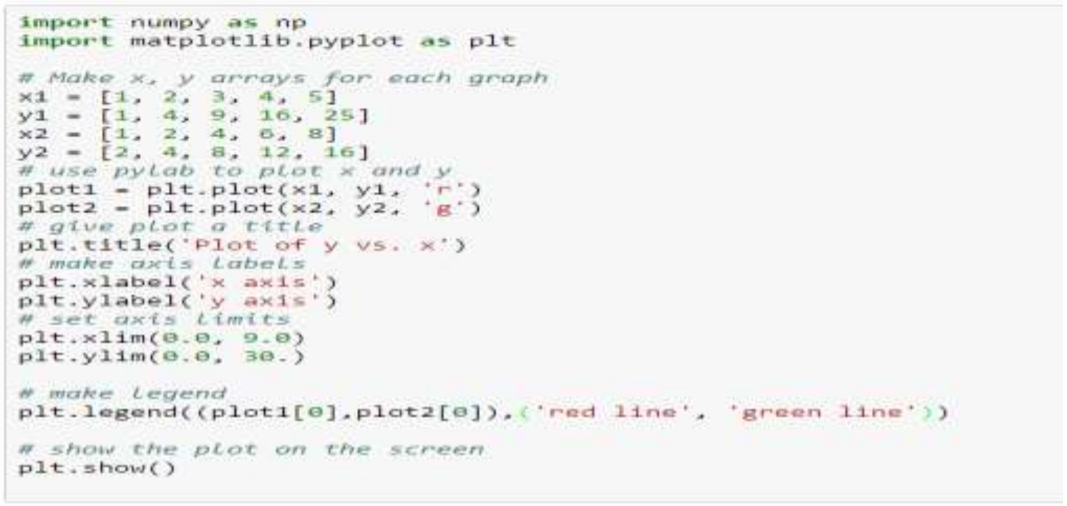
```
import numpy as np
import matplotlib.pyplot as plt
N = 5
menMeans = (20, 35, 30, 35, 27)
ind = np.arange(N)
print ind
plt.bar(ind, menMeans)
plt.title('Scores by group and gender')
plt.ylabel('Scores')
plt.xlabel('The number of people')
plt.legend( ('Men'))
plt.show()
```

```
[0 1 2 3 4]
```

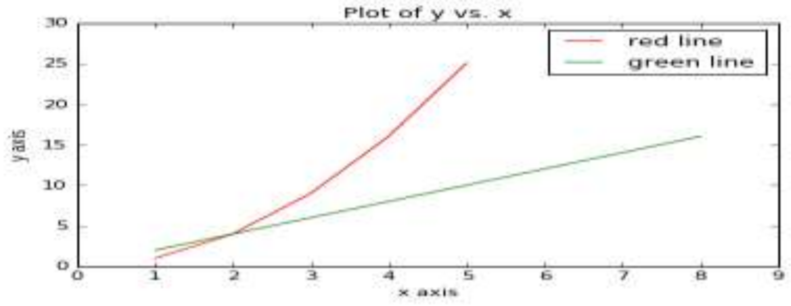


2개 범주

첫번째 파라미터에 plot처리 결과의 첫번째 요소, 두번째 파라미터에 label 처리

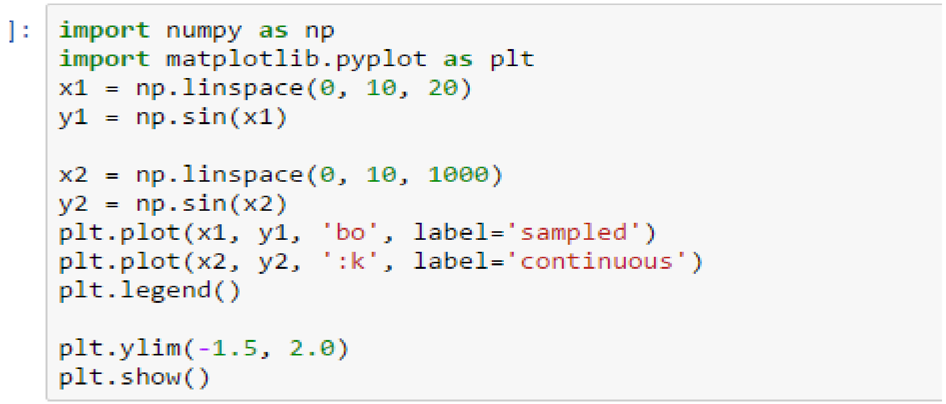


```
import numpy as np
import matplotlib.pyplot as plt

# Make x, y arrays for each graph
x1 = [1, 2, 3, 4, 5]
y1 = [1, 4, 9, 16, 25]
x2 = [1, 2, 4, 6, 8]
y2 = [2, 4, 8, 12, 16]
# use pylab to plot x and y
plot1 = plt.plot(x1, y1, 'r')
plot2 = plt.plot(x2, y2, 'g')
# give plot a title
plt.title('Plot of y vs. x')
# make axis labels
plt.xlabel('x axis')
plt.ylabel('y axis')
# set axis limits
plt.xlim(0.0, 9.0)
plt.ylim(0.0, 30.)

# make legend
plt.legend((plot1[0],plot2[0]),('red line', 'green line'))

# show the plot on the screen
plt.show()
```

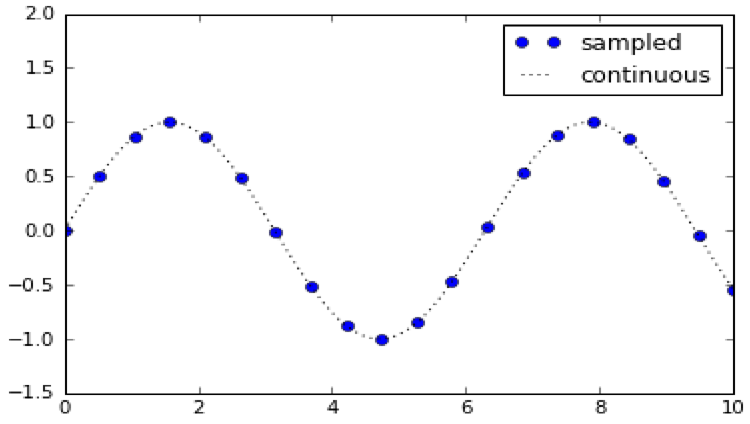


plot(label) 이용 : 1

plot함수의 label을 이용해서 그래프에 범주를 표시

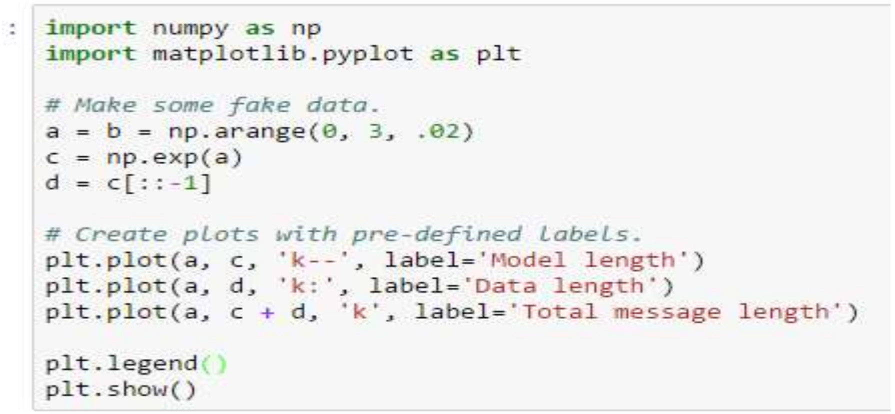


```
]: import numpy as np
   import matplotlib.pyplot as plt
   x1 = np.linspace(0, 10, 20)
   y1 = np.sin(x1)

   x2 = np.linspace(0, 10, 1000)
   y2 = np.sin(x2)
   plt.plot(x1, y1, 'bo', label='sampled')
   plt.plot(x2, y2, ':k', label='continuous')
   plt.legend()

   plt.ylim(-1.5, 2.0)
   plt.show()
```

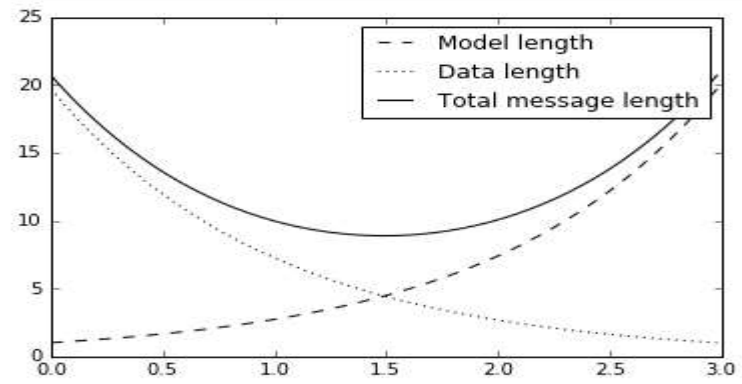


plot(label) 이용 : 2

plot함수의 label을 이용해서 그래프에 범주를 표시

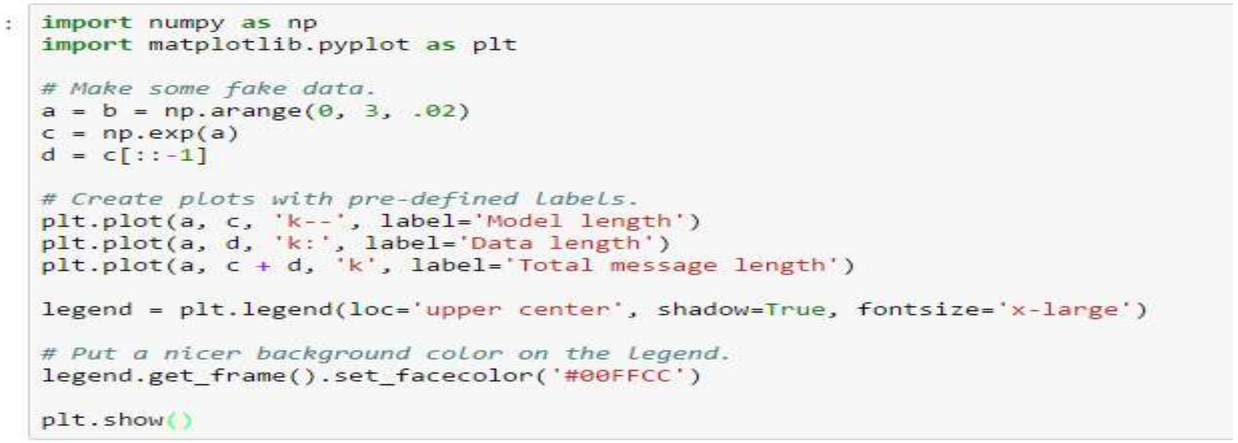


```
import numpy as np
import matplotlib.pyplot as plt

# Make some fake data.
a = b = np.arange(0, 3, .02)
c = np.exp(a)
d = c[::-1]

# Create plots with pre-defined labels.
plt.plot(a, c, 'k--', label='Model length')
plt.plot(a, d, 'k:', label='Data length')
plt.plot(a, c + d, 'k', label='Total message length')

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

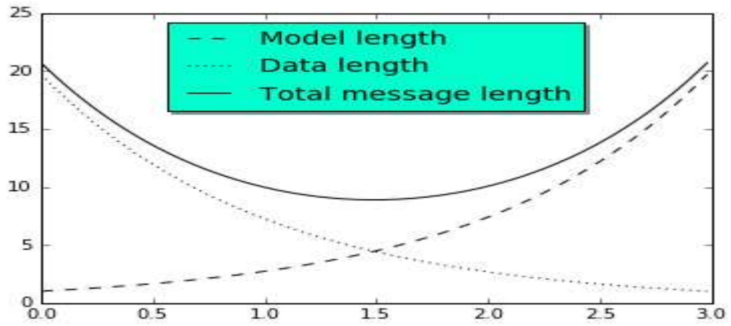


범주 위치 지정

legend 생성시 위치 배정 및 색깔 입히기



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

# Make some fake data.
a = b = np.arange(0, 3, .02)
c = np.exp(a)
d = c[::-1]

# Create plots with pre-defined labels.
plt.plot(a, c, 'k--', label='Model length')
plt.plot(a, d, 'k:', label='Data length')
plt.plot(a, c + d, 'k', label='Total message length')

legend = plt.legend(loc='upper center', shadow=True, fontsize='x-large')

# Put a nicer background color on the legend.
legend.get_frame().set_facecolor('#00FFCC')

plt.show()
```



MATPLOTLIB 파일처리

Moon Yong Joon

파일 처리 하기

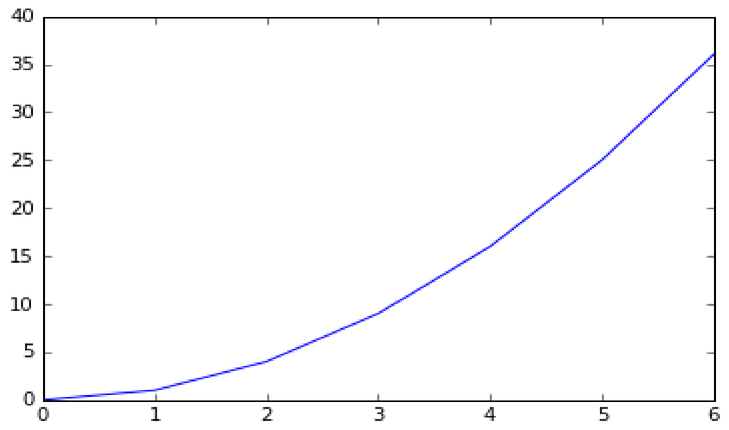
plot 함수 : file 읽고 처리

file를 읽고 Plot 함수를 통해 그래프 그리기

```
%%writefile data.txt
0 0
1 1
2 4
3 9
4 16
5 25
6 36
```

```
Writing data.txt
```

```
import matplotlib.pyplot as plt

X, Y = [] , []
for line in open('data.txt','r') :
    values = [float(s) for s in line.split()]
    X.append(values[0])
    Y.append(values[1])

plt.plot(X,Y)
plt.show()
```



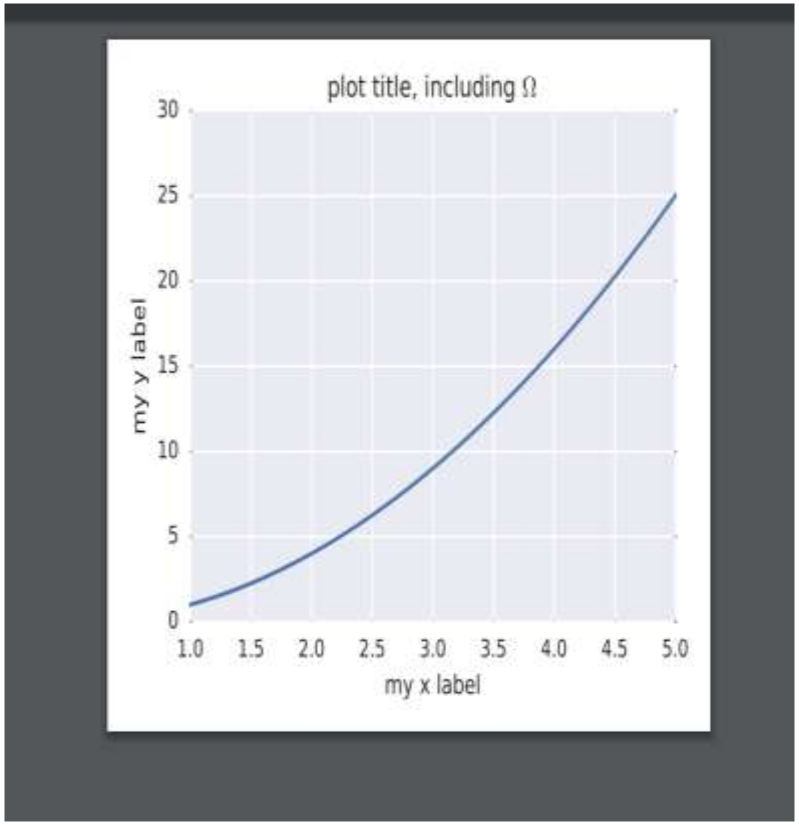
결과를 PDF 처리하기

savefig 함수

결과를 PDF로 보내기

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
f, ax = plt.subplots(1, 1, figsize=(5,4))
print f, ax
x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
ax.plot(x, y)
ax.set_xlim((1, 5))
ax.set_ylim((0, 30))
ax.set_xlabel('my x label')
ax.set_ylabel('my y label')
ax.set_title('plot title, including $\Omega$')
plt.tight_layout()
plt.savefig('line_plot_plus.pdf')
```

```
Figure(400x320) Axes(0.125,0.125;0.775x0.775)
```



DOCKER에서 JUPYTER NOTEBOK 설정

Moon Yong Joon

Windows 환경

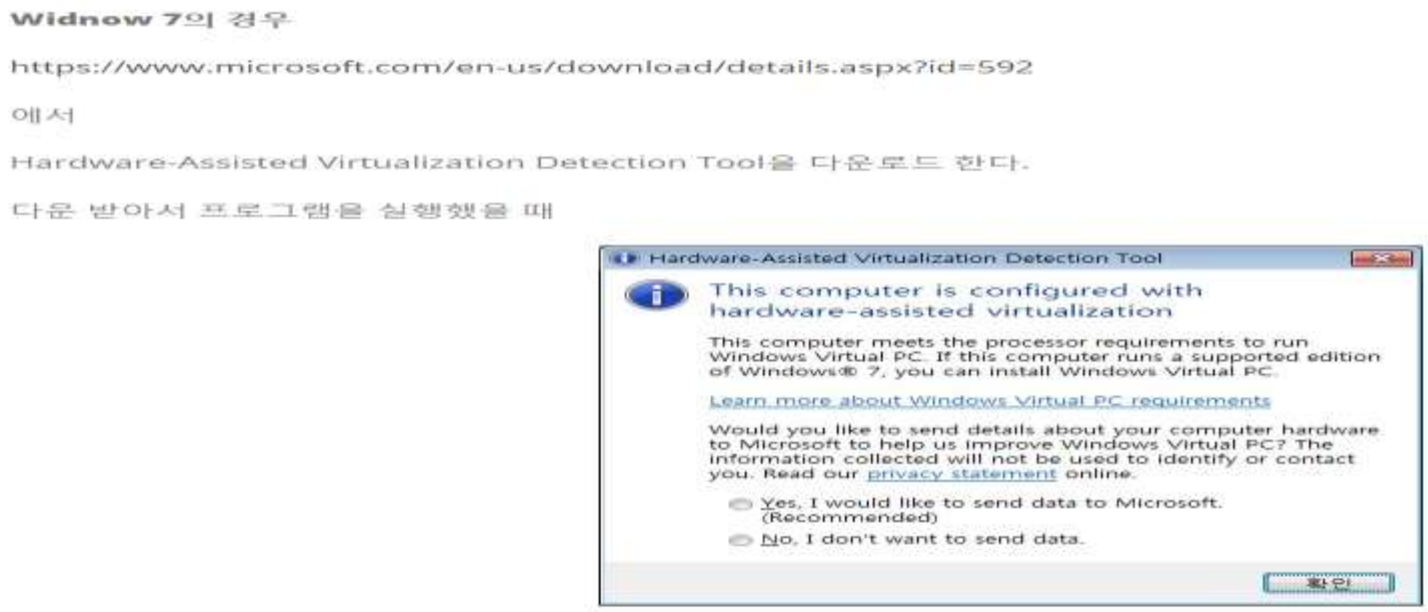
widndows 조건

windows내에서 docker를 사용하기 위한 조건

1. 64-bit 운영체제여야만 한다.
2. 윈도우 버전이 윈도우 7 혹은 그 이상이여야 한다.
3. OS에서 가상화(Virtualization)가 가능해야 한다.

widndows7 추가 업데이트

docker와 browser 연계를 위해서는 아래의 Tool을 설치



Widnow 7의 경우

https://www.microsoft.com/en-us/download/details.aspx?id=592

에서

Hardware-Assisted Virtualization Detection Tool을 다운로드 한다.

다운 받아서 프로그램을 실행했을 때

Hardware-Assisted Virtualization Detection Tool

This computer is configured with hardware-assisted virtualization

This computer meets the processor requirements to run Windows Virtual PC. If this computer runs a supported edition of Windows® 7, you can install Windows Virtual PC.

Learn more about Windows Virtual PC requirements

Would you like to send details about your computer hardware to Microsoft to help us improve Windows Virtual PC? The information collected will not be used to identify or contact you. Read our privacy statement online.

○ Yes, I would like to send data to Microsoft. (Recommended)

○ No, I don't want to send data.

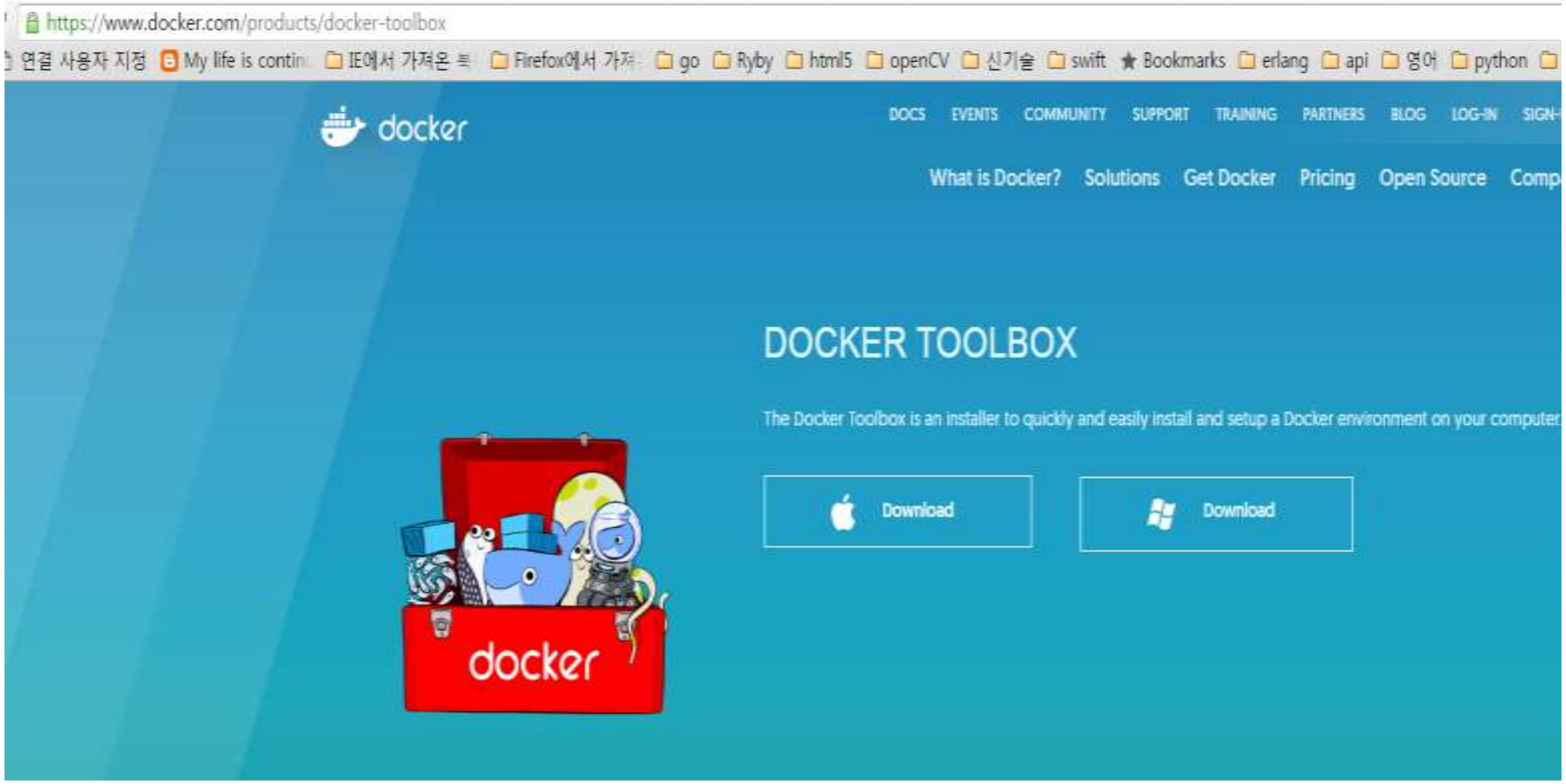
확인

Docker 설치

docker 다운로드 및 설치

docker설치시 virtualbox도 같이 설치

https://www.docker.com/products/docker-toolbox

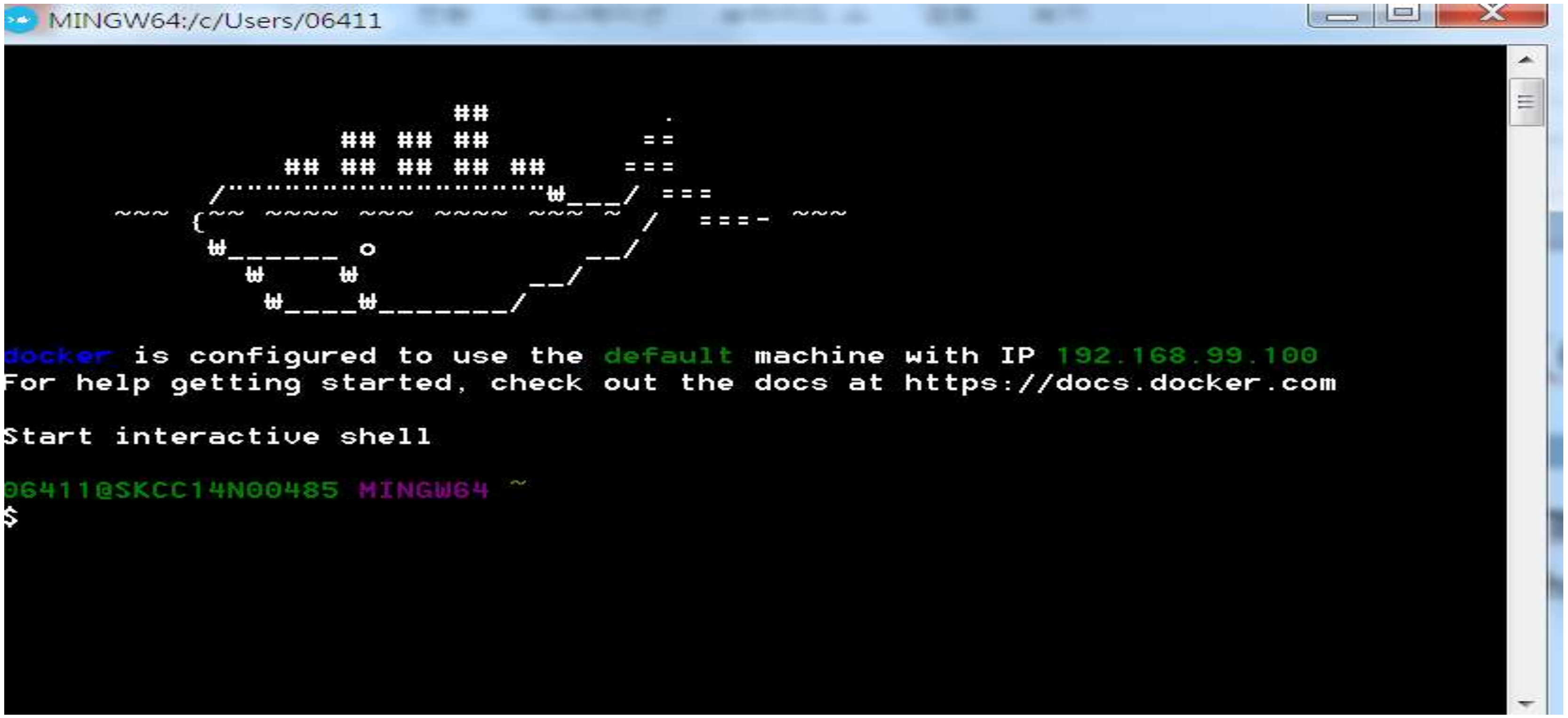


DOCKER TOOLBOX

The Docker Toolbox is an installer to quickly and easily install and setup a Docker environment on your computer

docker : 터미널 구동

docker 를 실행하기 Docker Quickstart Terminal 실행



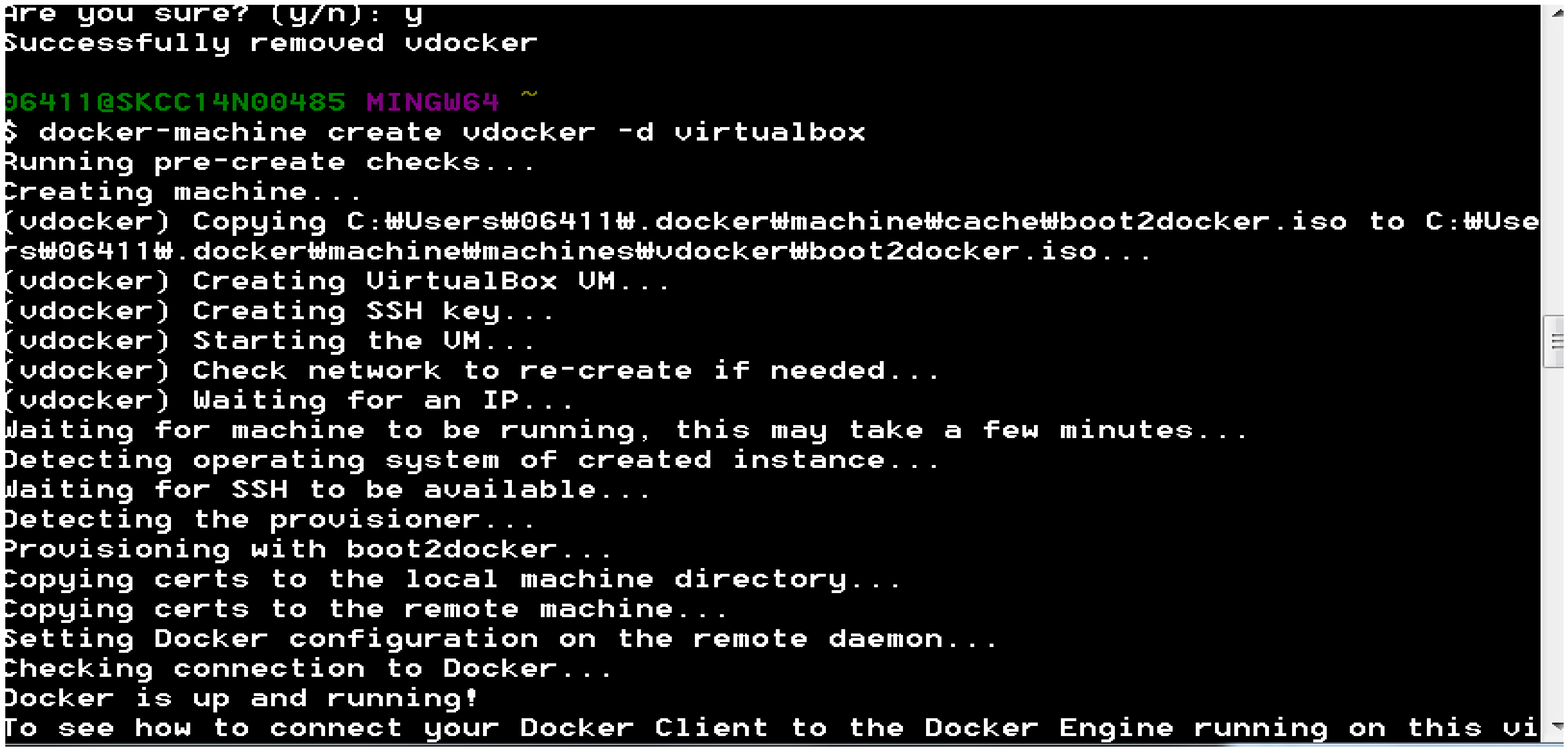
MINGW64:/c/Users/06411

```
                        ##         .
                  ## ## ##        ==
               ## ## ## ## ##    ===
           /"""""""""""""""""""₩___/ ===
      ~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ ~ /  ===- ~~~
           ₩______ o           __/
             ₩    ₩         __/
              ₩____₩_______/

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com

Start interactive shell

06411@SKCC14N00485 MINGW64 ~
$
```

docker : vdocker 만들기

docker-machine에 vdocker 만들기

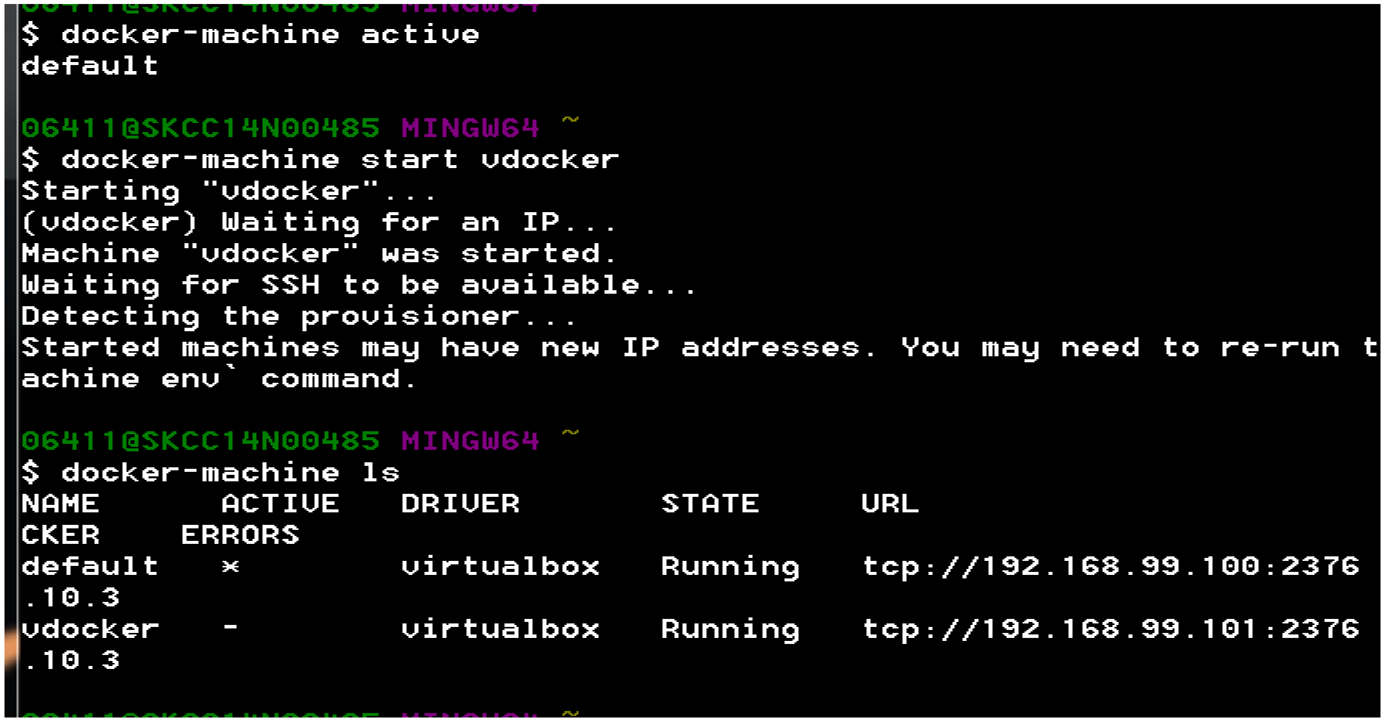
$ docker-machine create vdocker -d virtualbox



```
Are you sure? (y/n): y
Successfully removed vdocker

06411@SKCC14N00485 MINGW64 ~
$ docker-machine create vdocker -d virtualbox
Running pre-create checks...
Creating machine...
(vdocker) Copying C:₩Users₩06411₩.docker₩machine₩cache₩boot2docker.iso to C:₩Use
rs₩06411₩.docker₩machine₩machines₩vdocker₩boot2docker.iso...
(vdocker) Creating VirtualBox VM...
(vdocker) Creating SSH key...
(vdocker) Starting the VM...
(vdocker) Check network to re-create if needed...
(vdocker) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this vi
```

docker-machien 확인 방법

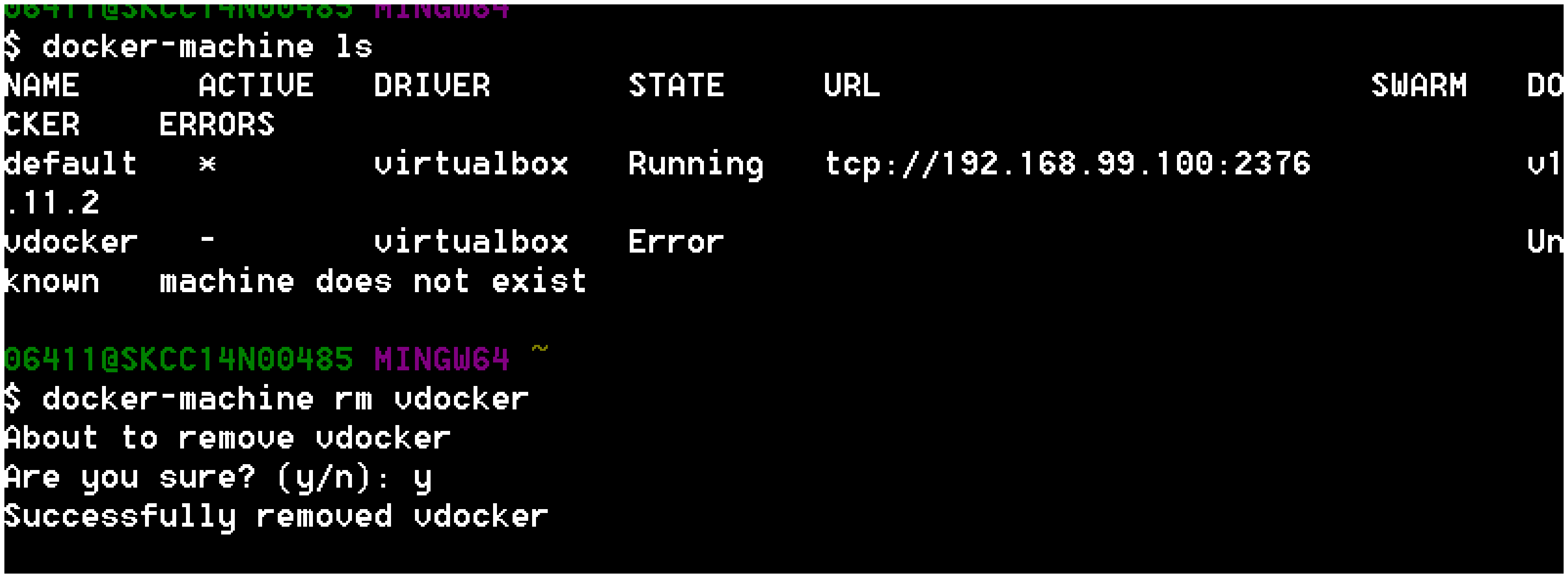
docker-machine을 확인하고 vdocker를 실행



```
$ docker-machine active
default

06411@SKCC14N00485 MINGW64 ~
$ docker-machine start vdocker
Starting "vdocker"...
(vdocker) Waiting for an IP...
Machine "vdocker" was started.
Waiting for SSH to be available...
Detecting the provisioner...
Started machines may have new IP addresses. You may need to re-run t
achine env` command.

06411@SKCC14N00485 MINGW64 ~
$ docker-machine ls
NAME        ACTIVE    DRIVER         STATE      URL
CKER      ERRORS
default     *         virtualbox     Running    tcp://192.168.99.100:2376
.10.3
vdocker     -         virtualbox     Running    tcp://192.168.99.101:2376
.10.3
```

docker-machien : error 처리

docker-machine을 확인하고 vdocker가 error 일 경우 rm 명령으로 삭제 후 재생성 필요



```
06411@SKCC14N00485 MINGW64
$ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DO
CKER    ERRORS
default   *        virtualbox   Running   tcp://192.168.99.100:2376           v1
.11.2
vdocker   -        virtualbox   Error                                         Un
known   machine does not exist

06411@SKCC14N00485 MINGW64 ~
$ docker-machine rm vdocker
About to remove vdocker
Are you sure? (y/n): y
Successfully removed vdocker
```

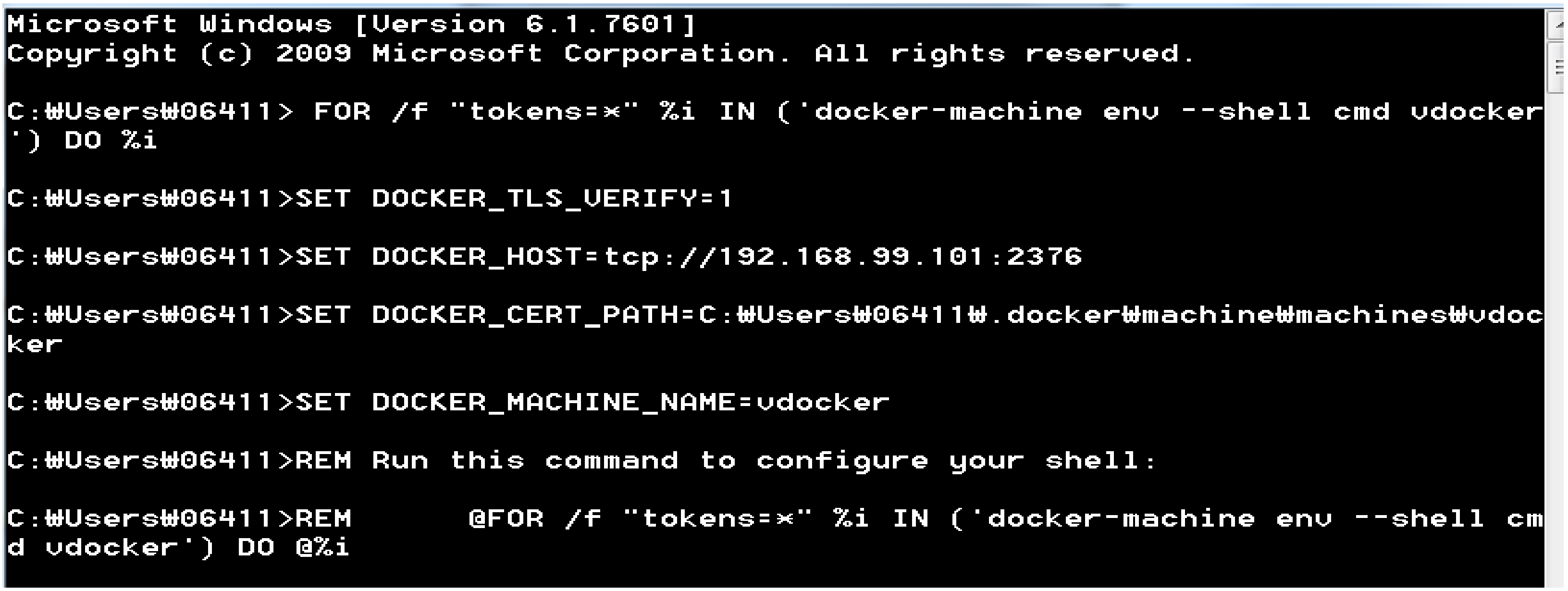
Docker 환경 세팅

windows :docker환경 세팅

윈도우 cmd prompt를 실행 (Docker Quickstart Terminal 창이 아님)

아래의 명령어를 입력한다.

FOR /f "tokens=*" %i IN ('docker-machine env --shell cmd vdocker') DO %i



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:₩Users₩06411> FOR /f "tokens=*" %i IN ('docker-machine env --shell cmd vdocker
') DO %i

C:₩Users₩06411>SET DOCKER_TLS_VERIFY=1

C:₩Users₩06411>SET DOCKER_HOST=tcp://192.168.99.101:2376

C:₩Users₩06411>SET DOCKER_CERT_PATH=C:₩Users₩06411₩.docker₩machine₩machines₩vdoc
ker

C:₩Users₩06411>SET DOCKER_MACHINE_NAME=vdocker

C:₩Users₩06411>REM Run this command to configure your shell:

C:₩Users₩06411>REM       @FOR /f "tokens=*" %i IN ('docker-machine env --shell cm
d vdocker') DO @%i
```

Docker 내에서 사용하기

windows :docker환경 세팅

Docker Quickstart Terminal 창에서
아래의 명령어를 입력한다.

```
docker run –it b.gcr.io/tensorflow/tensorflow:latest-devel
```

tensorflow의 설치가 완료되면 자동으로 리눅스 터미널 창으로 넘어간다.

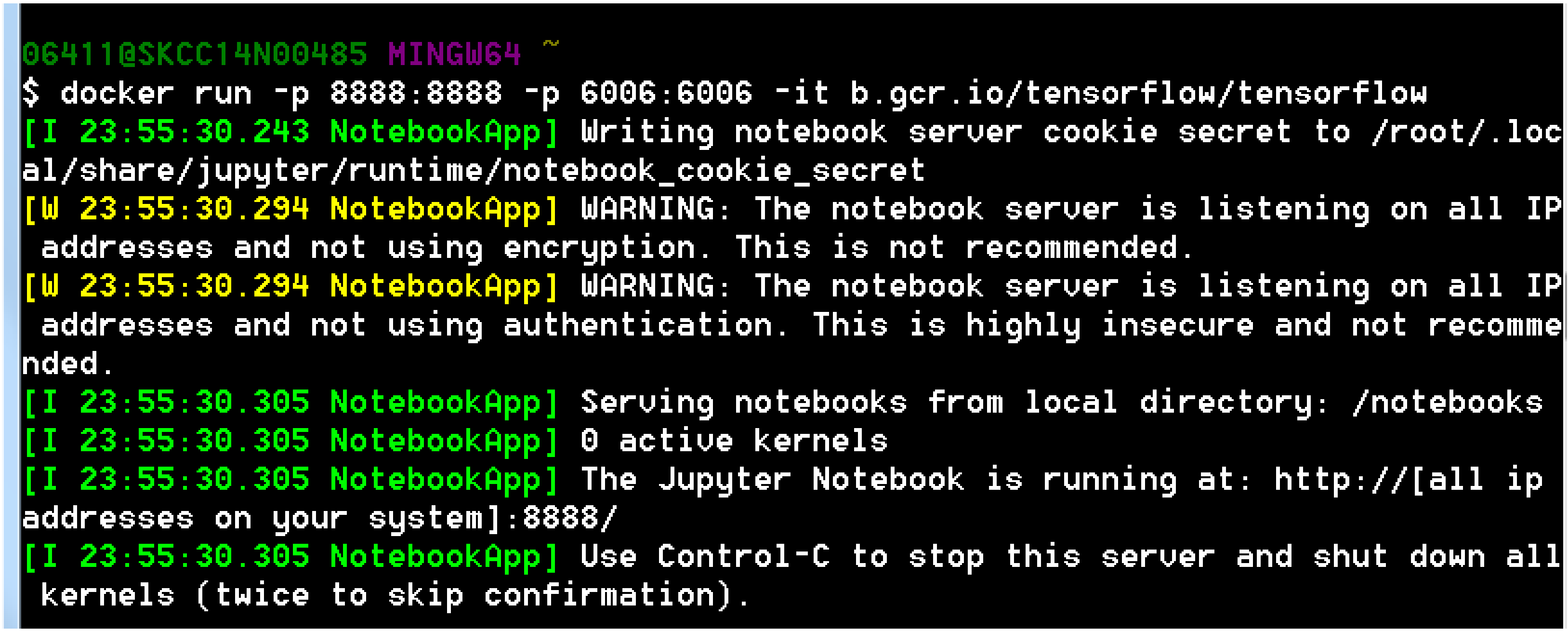
Jupyter notebook 사용하기

tensorflow 실행

docker terminal에서 아래의 명령을 실행

docker run –p 8888:8888 –p 6006:6006 –it b.gcr.io/tensorflow/tensorflow

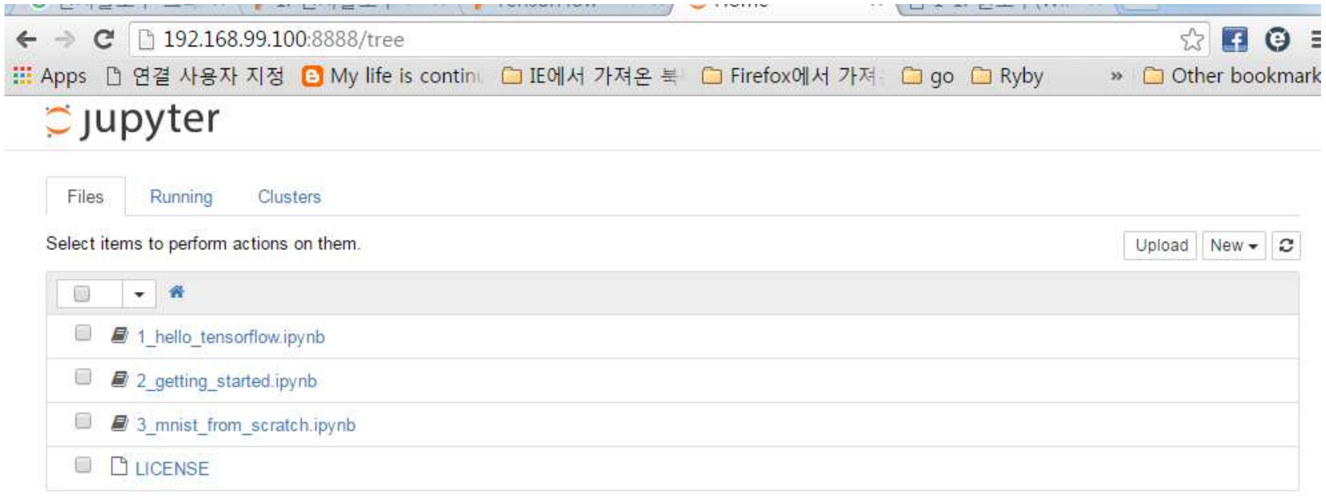
docker에서 8888(jupyter notebook), 6006(tensorboard)에 대해 구동



```
06411@SKCC14N00485 MINGW64 ~
$ docker run -p 8888:8888 -p 6006:6006 -it b.gcr.io/tensorflow/tensorflow
[I 23:55:30.243 NotebookApp] Writing notebook server cookie secret to /root/.loc
al/share/jupyter/runtime/notebook_cookie_secret
[W 23:55:30.294 NotebookApp] WARNING: The notebook server is listening on all IP
 addresses and not using encryption. This is not recommended.
[W 23:55:30.294 NotebookApp] WARNING: The notebook server is listening on all IP
 addresses and not using authentication. This is highly insecure and not recomme
nded.
[I 23:55:30.305 NotebookApp] Serving notebooks from local directory: /notebooks
[I 23:55:30.305 NotebookApp] 0 active kernels
[I 23:55:30.305 NotebookApp] The Jupyter Notebook is running at: http://[all ip
addresses on your system]:8888/
[I 23:55:30.305 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
```

jupyter notebook 실행

http://192.168.99.100:8888/ 을 웹브라우저에 입력하면 실행됨


192.168.99.100:8888/tree

Select items to perform actions on them.

1_hello_tensorflow.ipynb
2_getting_started.ipynb
3_mnist_from_scratch.ipynb
LICENSE

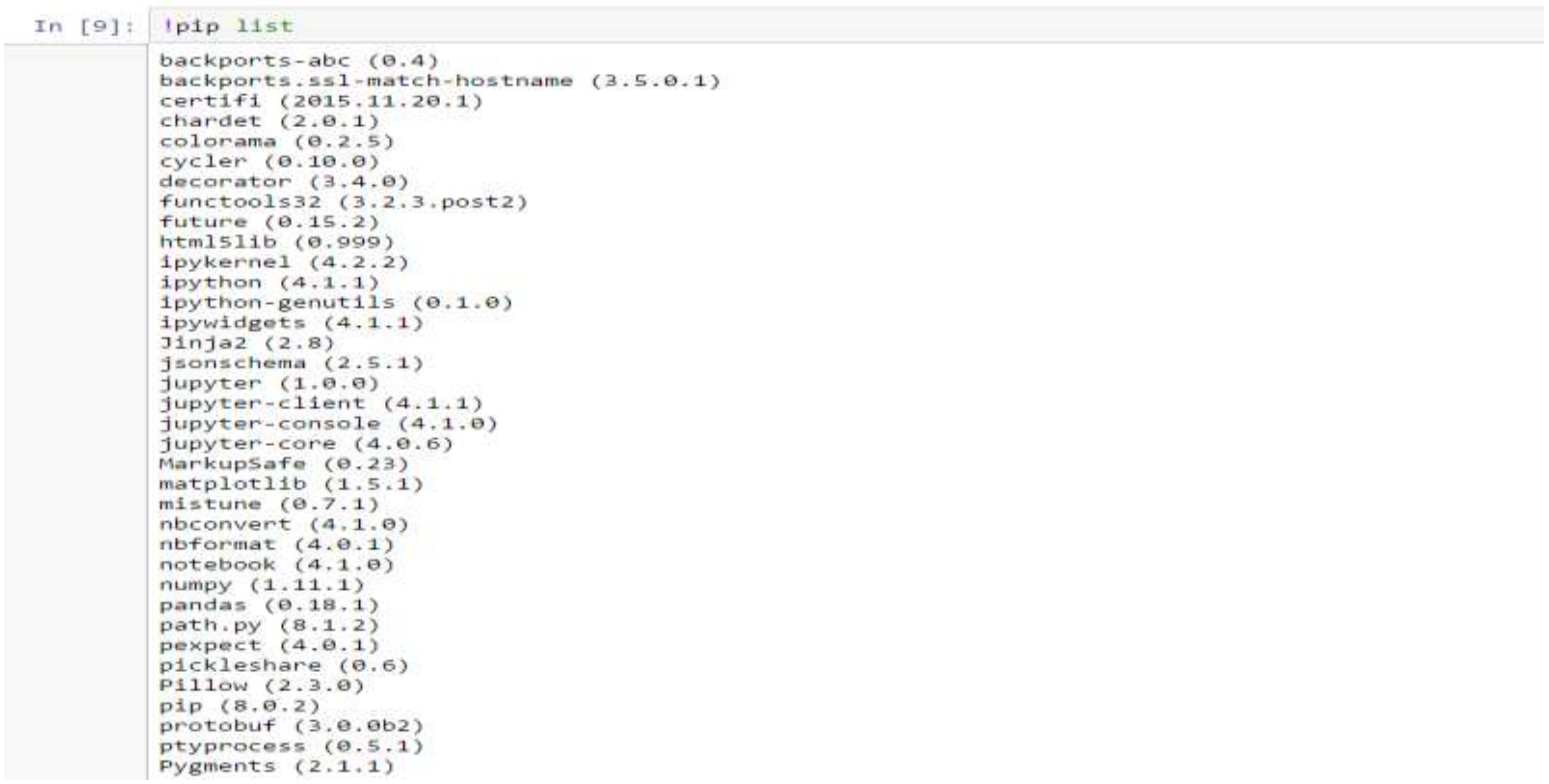
PYTHON MODULE INSTALL/ UPGRADE

Moon Yong Joon

Python 모듈 추가하기

Jupyter notebook 에서 모듈조회

!pip list로 모듈 조회(docker 사용시는 docker 이미지 내의 python 모듈 리스트가 조회됨)



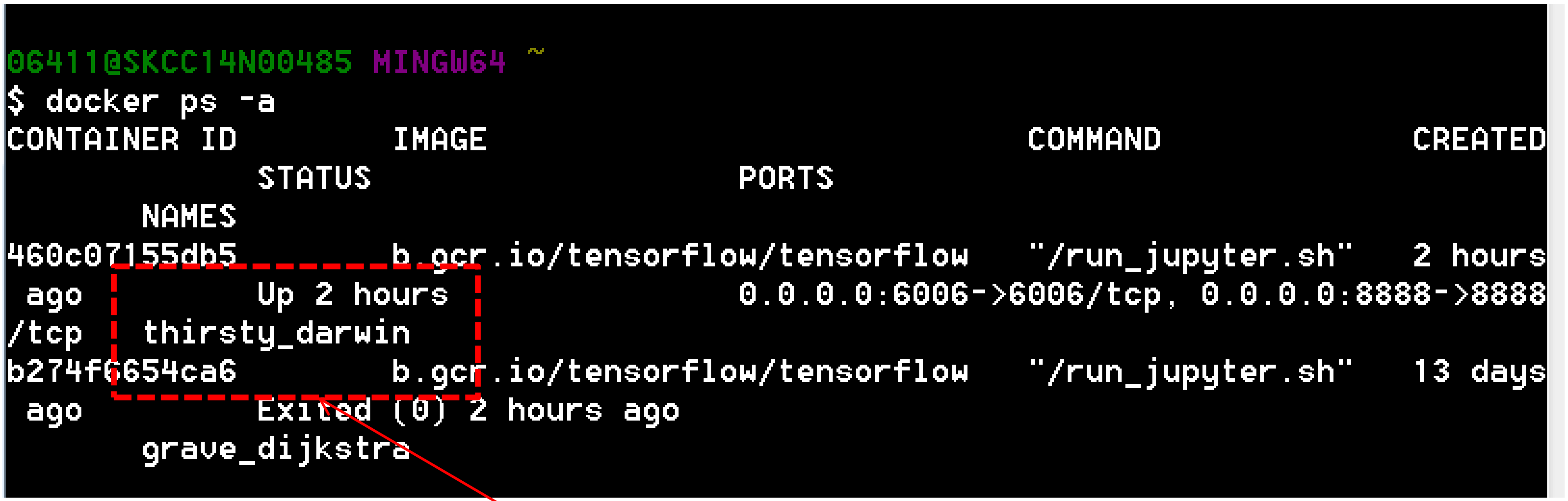
```
In [9]: !pip list
```

```
backports-abc (0.4)
backports.ssl-match-hostname (3.5.0.1)
certifi (2015.11.20.1)
chardet (2.0.1)
colorama (0.2.5)
cycler (0.10.0)
decorator (3.4.0)
functools32 (3.2.3.post2)
future (0.15.2)
html5lib (0.999)
ipykernel (4.2.2)
ipython (4.1.1)
ipython-genutils (0.1.0)
ipywidgets (4.1.1)
Jinja2 (2.8)
jsonschema (2.5.1)
jupyter (1.0.0)
jupyter-client (4.1.1)
jupyter-console (4.1.0)
jupyter-core (4.0.6)
MarkupSafe (0.23)
matplotlib (1.5.1)
mistune (0.7.1)
nbconvert (4.1.0)
nbformat (4.0.1)
notebook (4.1.0)
numpy (1.11.1)
pandas (0.18.1)
path.py (8.1.2)
pexpect (4.0.1)
pickleshare (0.6)
Pillow (2.3.0)
pip (8.0.2)
protobuf (3.0.0b2)
ptyprocess (0.5.1)
Pygments (2.1.1)
```

Python 모듈 추가하기

Docker 명령으로 컨테이너 확인

docker ps –a 명령으로 현재 컨테이너 확인

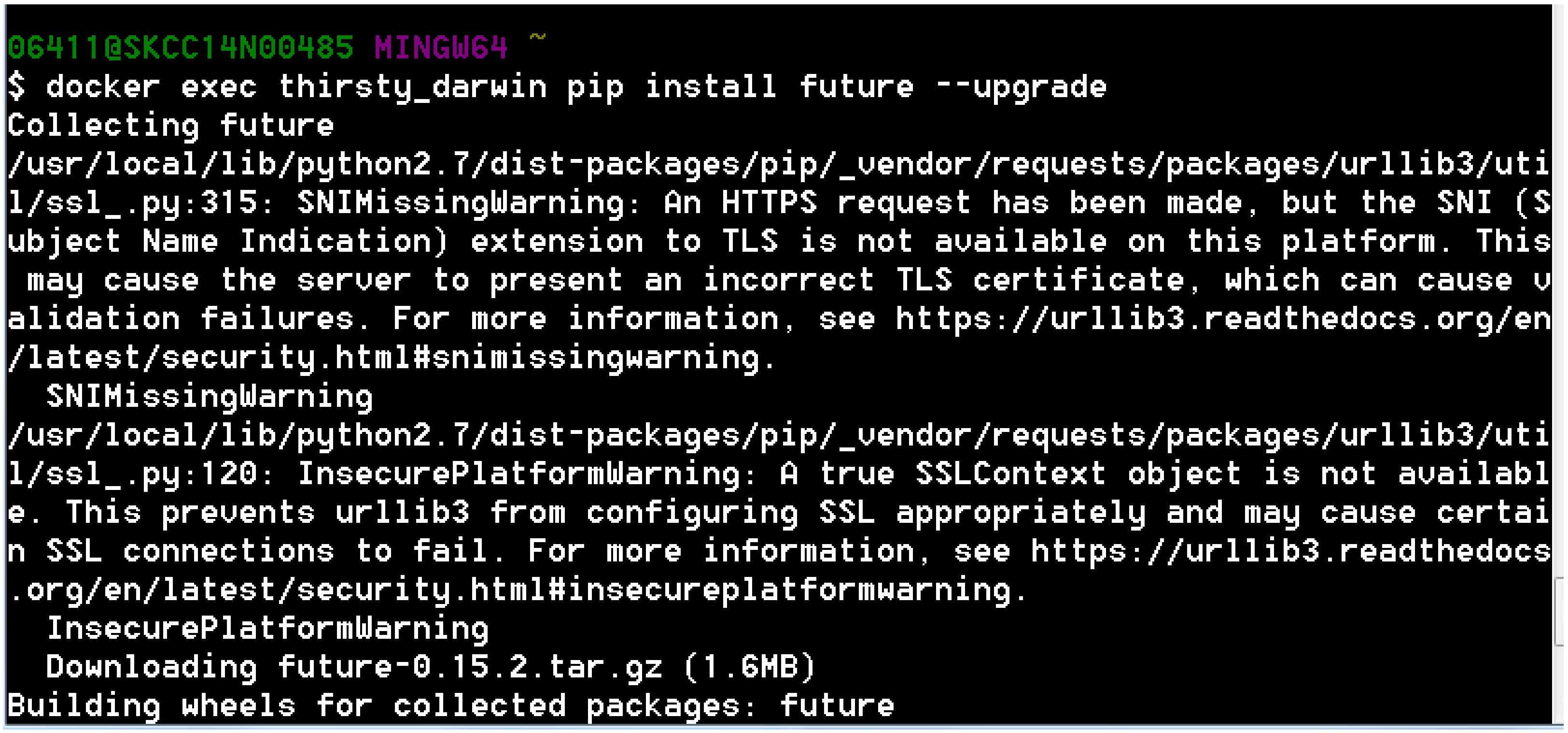


```
06411@SKCC14N00485 MINGW64 ~
$ docker ps -a
CONTAINER ID        IMAGE                              COMMAND               CREATED
             STATUS                        PORTS
       NAMES
460c07155db5        b.gcr.io/tensorflow/tensorflow     "/run_jupyter.sh"     2 hours
 ago                Up 2 hours                    0.0.0.0:6006->6006/tcp, 0.0.0.0:8888->8888
/tcp    thirsty_darwin
b274f6654ca6        b.gcr.io/tensorflow/tensorflow     "/run_jupyter.sh"     13 days
 ago                Exited (0) 2 hours ago
       grave_dijkstra
```

현재 실행중인 컨테이너 명

Docker 명령으로 모듈 추가

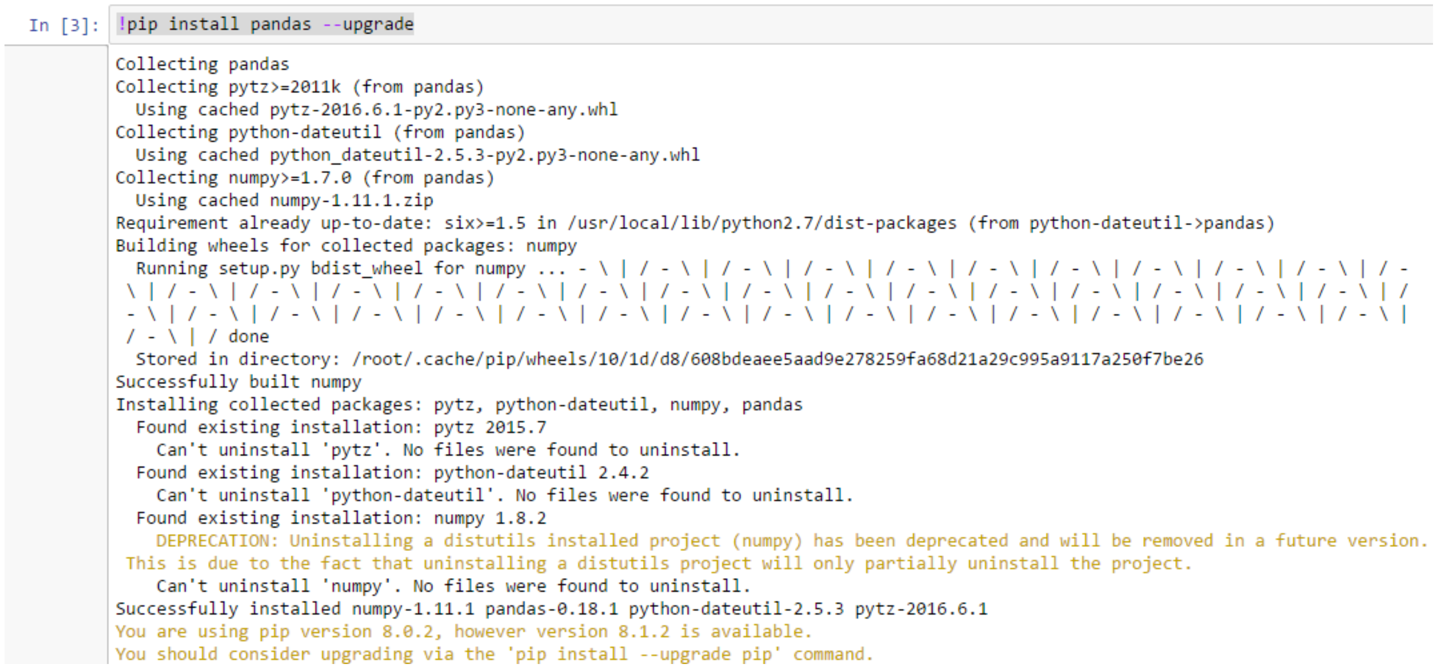
docker exec {컨테이너명} {pip 명령} 으로 실행



```
06411@SKCC14N00485 MINGW64 ~
$ docker exec thirsty_darwin pip install future --upgrade
Collecting future
/usr/local/lib/python2.7/dist-packages/pip/_vendor/requests/packages/urllib3/uti
l/ssl_.py:315: SNIMissingWarning: An HTTPS request has been made, but the SNI (S
ubject Name Indication) extension to TLS is not available on this platform. This
 may cause the server to present an incorrect TLS certificate, which can cause v
alidation failures. For more information, see https://urllib3.readthedocs.org/en
/latest/security.html#snimissingwarning.
  SNIMissingWarning
/usr/local/lib/python2.7/dist-packages/pip/_vendor/requests/packages/urllib3/uti
l/ssl_.py:120: InsecurePlatformWarning: A true SSLContext object is not availabl
e. This prevents urllib3 from configuring SSL appropriately and may cause certai
n SSL connections to fail. For more information, see https://urllib3.readthedocs
.org/en/latest/security.html#insecureplatformwarning.
  InsecurePlatformWarning
  Downloading future-0.15.2.tar.gz (1.6MB)
Building wheels for collected packages: future
```

Jupyter notebook 에서 모듈 추가

!pip install {모듈명} –upgrade 로 추가



```
In [3]: !pip install pandas --upgrade

Collecting pandas
Collecting pytz>=2011k (from pandas)
  Using cached pytz-2016.6.1-py2.py3-none-any.whl
Collecting python-dateutil (from pandas)
  Using cached python_dateutil-2.5.3-py2.py3-none-any.whl
Collecting numpy>=1.7.0 (from pandas)
  Using cached numpy-1.11.1.zip
Requirement already up-to-date: six>=1.5 in /usr/local/lib/python2.7/dist-packages (from python-dateutil->pandas)
Building wheels for collected packages: numpy
  Running setup.py bdist_wheel for numpy ... - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / -
 \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | /
 - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ | / - \ |
 / - \ | / done
  Stored in directory: /root/.cache/pip/wheels/10/1d/d8/608bdeaee5aad9e278259fa68d21a29c995a9117a250f7be26
Successfully built numpy
Installing collected packages: pytz, python-dateutil, numpy, pandas
  Found existing installation: pytz 2015.7
    Can't uninstall 'pytz'. No files were found to uninstall.
  Found existing installation: python-dateutil 2.4.2
    Can't uninstall 'python-dateutil'. No files were found to uninstall.
  Found existing installation: numpy 1.8.2
    DEPRECATION: Uninstalling a distutils installed project (numpy) has been deprecated and will be removed in a future version.
 This is due to the fact that uninstalling a distutils project will only partially uninstall the project.
    Can't uninstall 'numpy'. No files were found to uninstall.
Successfully installed numpy-1.11.1 pandas-0.18.1 python-dateutil-2.5.3 pytz-2016.6.1
You are using pip version 8.0.2, however version 8.1.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```