# ALPS チュートリアル: Introduction to Matplotlib

CMSI 神戸ハンズオン

ALPS Collaboration
http://alps.comp-phys.org/

# デフォルトの環境設定

図の中で使われる文字のサイズや，図の縦・横のサイズなどをあらかじめ設定しておくことができます．

- matplotlib ${HOME}/.matplotlib/matplotlibrc で設定する
  - テンプレを $PYTHONHOME/lib/site-packages/matplotlib/mpl-data/matplotlibrc からコピーして使う
  - プロット時にウィンドウを開いて図を表示するには，Mac OS X の場合は次のような設定が必要

```
#backend : Agg
backend : MacOSX
```

  - 逆にウィンドウを開きたくないなら Agg に設定する
  - AGG: Anti-Grain Geometry. ベクトルデータを画像に描くライブラリ．C++ 用．

# 動的に環境設定を行う

- backend の設定は import matplotlib の直後にすること

```
>>> import matplotlib as mpl
>>> mpl.use('Agg')
```

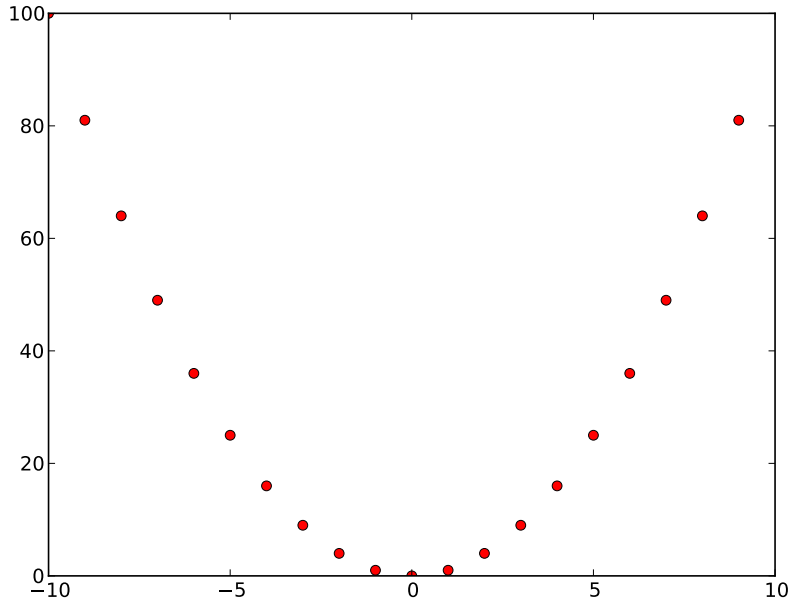
- グラフのサイズやフォントのサイズの設定

```
>>> import pylab
>>> params = {
...     'legend.fontsize': 10
...     'xtick.labelsize':  8
...     'ytick.labelsize':  8
...     'figure.figsize':  [width, height]}
>>> pyalps.rcParams.update(params)
```

# 簡単な図をプロットしてみる

## $y = x^2$ のプロット

```
>>> x = range(-10,10)
>>> y = [i*i for i in x]
# シンボルの色を赤(r),
# 形を丸(o)に設定
>>> pl.plot(x, y, 'ro')
...[<matplotlib.lines.Line2D
    at 0x114c82710>]
# 拡張子で保存の
# ファイル形式が決まります!
>>> pl.savefig('x2.pdf')
>>> pl.show()
```



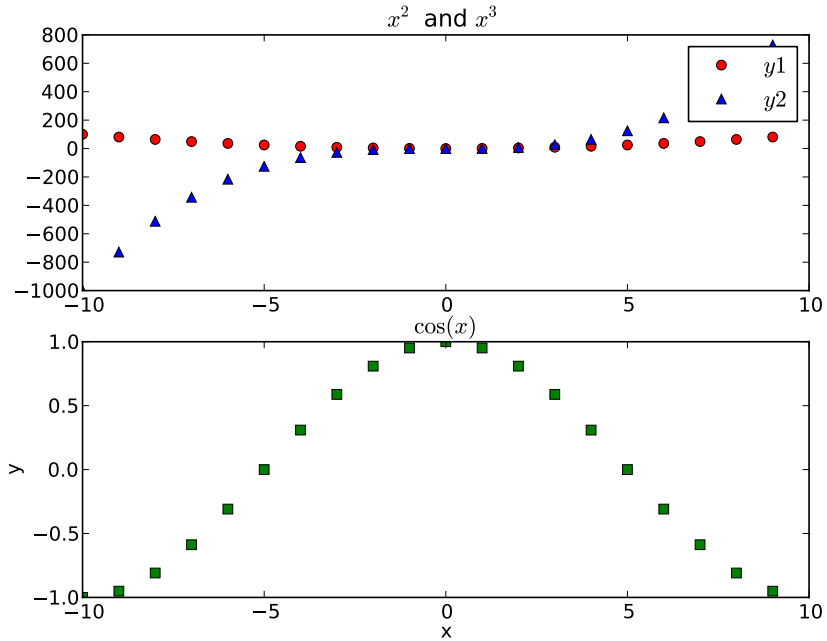
# 少し複雑な図をプロットしてみる

```
>>> import math as m          # cos を使うため
>>> x = range(-10,10)
>>> y1 = [i*i for i in x]
>>> y2 = [i*i*i for i in x]
>>> y3 = [m.cos(2 * m.pi * i/20) for i in x]
# 図を上下 2 段に並べてプロットします
>>> pl.subplot(211)           # 上側の図を書き始めます
>>> pl.plot(x, y1, 'ro')      # 1 つのグラフに
>>> pl.plot(x, y2, 'b^')      # 2 つのデータセットをプロ
    ット
>>> pl.title('$x^2$␣and␣$x^3$')   # 図のタイトル. TeX
    も Ok!
>>> pl.legend(('$y1$', '$y2$'), numpoints=1)  # 凡例
```

# 続き

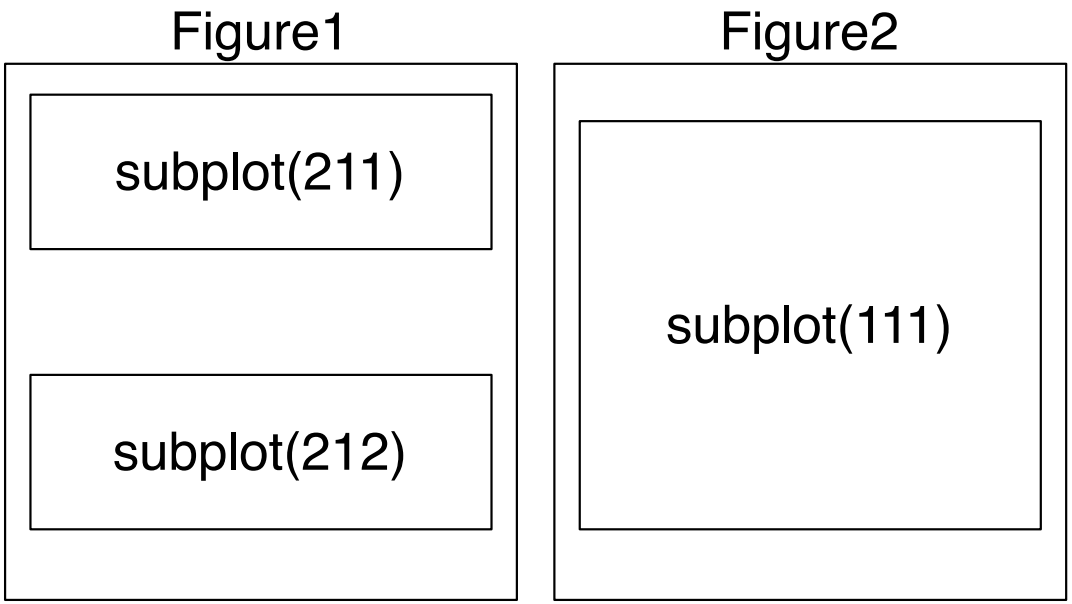
```
# 下側の図を書き始めます
>>> pl.subplot(212)
>>> pl.plot(x, y3, 'gs')
>>> pl.title('$\cos(x)$')
>>> pl.xlabel('x')
>>> pl.ylabel('y')
>>> pl.savefig('x3.pdf')
>>> pl.show()
```

出来上がり!



# Figure と Axes

Matplotlib の描画領域には 2 つの座標系があります．Figure と Axes です．1 つの Figure は 1 枚の紙で，そこに複数のグラフ (Axes) を描くというイメージです．
Subplot は Axes の特殊なものです．subplot(col,row,pos) で描画位置を指定できます．



# Current Figure/Axes vs. オブジェクト指向

今まで紹介してきたプロットは，暗黙のうちに Current Figure/Axes への操作を行っていました．複数の Figure を同時に扱いたい場合はオブジェクト指向的なプロット方法が便利です．

```
import pylab as pl

pl.figure()
pl.plot(pl.rand(1000), 'o')

pl.figure()
pl.plot(pl.rand(1000), '^')

pl.show()
```

```
import pylab as pl
fig1 = pl.figure()
fig2 = pl.figure()

ax1 = fig1.add_subplot(111)
ax2 = fig2.add_subplot(111)

ax1.plot(pl.rand(1000), 'o')
ax2.plot(pl.rand(1000), '^')

pl.show()
```