شماره1

'''

This illustrates the NonUniformImage class, which still needs

an axes method interface; either a separate interface, or a

generalization of imshow.

'''

from matplotlib.pyplot import figure, show

import numpy as np

from matplotlib.image import NonUniformImage

interp='nearest'

x = np.linspace(-4, 4, 9)

x2 = x\*\*3

y = np.linspace(-4, 4, 9)

#print 'Size %d points' % (len(x) \* len(y))

z = np.sqrt(x[np.newaxis,:]\*\*2 + y[:,np.newaxis]\*\*2)

fig = figure()

fig.suptitle('NonUniformImage class')

ax = fig.add\_subplot(221)

im = NonUniformImage(ax, interpolation=interp, extent=(-4,4,-4,4))

im.set\_data(x, y, z)

ax.images.append(im)

ax.set\_xlim(-4,4)

ax.set\_ylim(-4,4)

ax.set\_title(interp)

ax = fig.add\_subplot(222)

im = NonUniformImage(ax, interpolation=interp, extent=(-64,64,-4,4))

im.set\_data(x2, y, z)

ax.images.append(im)

ax.set\_xlim(-64,64)

ax.set\_ylim(-4,4)

ax.set\_title(interp)

interp = 'bilinear'

ax = fig.add\_subplot(223)

im = NonUniformImage(ax, interpolation=interp, extent=(-4,4,-4,4))

im.set\_data(x, y, z)

ax.images.append(im)

ax.set\_xlim(-4,4)

ax.set\_ylim(-4,4)

ax.set\_title(interp)

ax = fig.add\_subplot(224)

im = NonUniformImage(ax, interpolation=interp, extent=(-64,64,-4,4))

im.set\_data(x2, y, z)

ax.images.append(im)

ax.set\_xlim(-64,64)

ax.set\_ylim(-4,4)

ax.set\_title(interp)

show()

شماره2

#!/usr/bin/env python

"""

You can specify whether images should be plotted with the array origin

x[0,0] in the upper left or upper right by using the origin parameter.

You can also control the default be setting image.origin in your

matplotlibrc file; see http://matplotlib.sourceforge.net/matplotlibrc

"""

from pylab import \*

x = arange(100.0); x.shape = 10,10

interp = 'bilinear';

#interp = 'nearest';

lim = -2,11,-2,6

subplot(211, axisbg='g')

title('blue should be up')

imshow(x, origin='upper', interpolation=interp)

#axis(lim)

subplot(212, axisbg='y')

title('blue should be down')

imshow(x, origin='lower', interpolation=interp)

#axis(lim)

show()