# Matplotlib架构



## backend layer.



User interface backends (interactive)

1. GTK 2.x and GTK 3.x
2. wxWidgets
3. Tk
4. Qt4 and Qt5
5. Mac OS X Cocoa

Hardcopy backends (noninteractive)

1. PS
2. PDF
3. SVG
4. PNG

Furthermore, the user-interface and hardcopy backends are built upon some core abstractions. The base classes for these are as follows:

1. FigureCanvasBase and FigureManagerBase
2. RendererBase and GraphicsContextBase
3. Event, ShowBase, and Timer

**FigureCanvasBase**

The FigureCanvasBase class is a base class that is used by the user interface and hardcopy backends. It represents the canvas in which the Figure will render. Its responsibilities include the following:

1. Holding a reference to the Figure
2. Updating the Figure with a reference to the canvas
3. Defining event methods that run registered
4. Translating native toolkit events into the matplotlib event abstraction framework
5. Defining draw methods to render the Figure
6. Methods to start and stop non-GUI event loops

**RendererBase**

In matplotlib, the renderer handles the drawing operations. RendererBase was originally inspired by the GIMP drawing toolkit's Drawable class, and this is evident when one examines its drawing methods to render paths, images, Gouraud triangles, text, markers, path collections, and quad meshes.

Note that many of the render operations are handed off to an additional abstraction—GraphicsContextBase. This abstraction provides a clean separation for code that handles color, line styles, hatching styles, blending properties, and antialiasing options, among others.

**Event**

There are several aspects of the matplotlib backend that have to do with events, event loops, and timing. These responsibilities are divided across three base classes:

1. Event: This is the base class for DrawEvent, MouseEvent, and KeyEvent, among others
2. ShowBase: This is subclassed at the module level in the GUI backends
3. TimerBase: This is the base class for TimerQT, TimerGTK3, and TimerWx, to name a few

## artist layer



The artist layer constitutes the bulk of what matplotlib actually does—the generation of the plots for the purpose of display, manipulation, and publication. Most work in the artist layer is performed by a number of classes, most of which are derived from the Artist base class.

The artist layer is concerned with things such as the lines, shapes, axes, text, and so on. These are the subclasses of the Artist class that define things such as the following:

1. A canvas-artist coordinate transformation
2. Visibility
3. A clip box that defines the paintable area
4. Labels
5. A callback registry instance to handle user interaction events

The Artist subclasses can be classified into one of the following two groups:

1. Primitives (原语)
2. Containers

The following two sections provide more details about these groups.

## scripting layer



The scripting layer is the user-facing interface that simplifies the task of working with other layers

Here are the important points that you need to know about pyplot:

1. When imported, pyplot selects either the default backend for your system, or the one that you have previously configured
   1. After selecting a backend, pyplot calls a setup function which does the following: Creates a figure manager factory function, which when called will create a new figure manager appropriate for the selected backend
   2. Prepares the drawing function that should be used with the selected backend (taking into account whether the backend is a hardcopy or a user interface)
   3. Identifies the callable function that integrates with the backend mainloop function
   4. Provides the module for the selected backend

The pyplot interface defines a series of functions that depend on the components returned by the setup function. These include the following functions:

1. plot(): This function calls the plot method in the current figure's Axes object and the figure canvas's draw\* method (as identified in the preceding setup)
2. title(): This function sets the title of the current figure's Axes instance
3. savefig(): This function saves the current figure
4. draw(): This function redraws the current figure
5. gcf(): This function returns the current figure
6. gca(): This function returns the Axes instance of the current figure
7. get\_current\_fig\_manager(): This returns the current figure manager
8. figure(): This is a Figure factory function
9. switch\_backend(): This is a function that lets one easily change the selected backend

## The supporting components of the matplotlib stack

In addition to the three major components of the matplotlib stack, there are supporting components. These include the following:

1. **Configuration support**
2. **Utility modules and functions**
3. **C extensions**
4. **External libraries upon which matplotlib depends**



# 颜色、线条、标记

Code:

plot(randn(30).cumsum(), color='k', linestyle='dashed', marker='o')

## 颜色

### 颜色表示方式

#### 方式一：rgb

1)(r,g,b) r,g,b范围[0,1]

举例：

line1 = [1,2,3,4,5]

plt.plot(line1, color=(0,0,1))

2)#rrggbb rr,gg,bb用16进制表示,范围为00~ff

举例：

line1 = [1,2,3,4,5]

plt.plot(line1, color='#0000ff')

#### 方式二：rgba

1. (r,g,b,a)其中r,g,b与方式一种一致，a表示透明度

举例：

line2 = [5,4,3,2,1]

plt.plot(line3, color=(0,0,1,0.3))

2)#rrggbbaa rr,gg,bb,aa用16进制表示,范围为00~ff

#### 方式三: 预定义好的名称

举例：

line3 = [3,3,3,3,3]

plt.plot(line3, color='r')

预定义好的名称有：

*========== ========*

*character color*

*========== ========*

*'b' blue*

***'g'*** *green*

*'r' red*

*'c' cyan*

*'m' magenta*

*'y' yellow*

*'k' black*

*'w' white*

*========== ========*

*或者1)*

*Hex strings (``'#008000'``)*

#### 方式四: 灰度

灰度用字符表示，字符内为0.0~1.0之间的数值。

举例：

line1 = [1,2,3,4,5]

plt.plot(line1, color='0.75')

### edge color

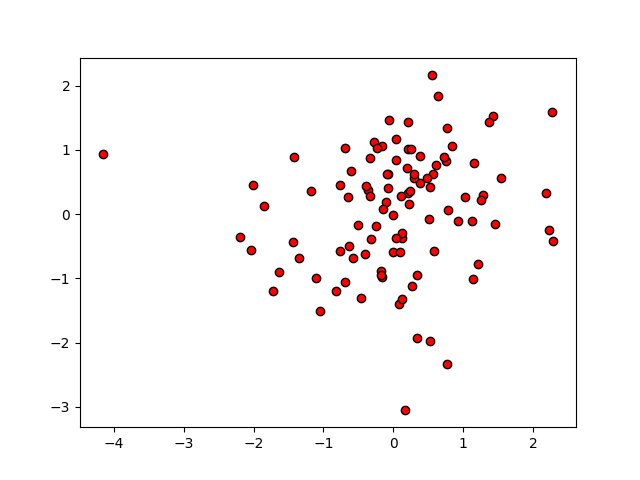
**color** parameter controls the color of the dots, the **edgecolor** parameter controls the color of the edge of the dots

举例：

data = np.random.standard\_normal((100, 2))

plt.scatter(data[:,0], data[:,1], color = 'r', edgecolor='0.0')

输出：



### 色图（colormaps）

When using a lot of colors, defining each color one by one is tedious. Moreover, building a good set of colors is a problem in itself. In some cases, **colormaps** can address those issues. Colormaps define colors with a continuous function of one variable to one value, corresponding to one color. matplotlib provides several common colormaps; most of them are continuous color ramps. In this recipe, we are going to see how to color scatter plots with a colormap.

Code:

import numpy as np

import matplotlib.cm as cm

import matplotlib.pyplot as plt

N = 256

angle = np.linspace(0, 8 \* 2 \* np.pi, N)

radius = np.linspace(.5, 1., N)

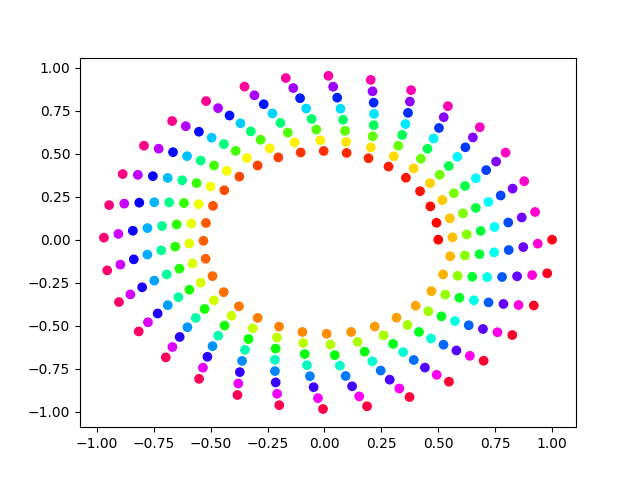
X = radius \* np.cos(angle)

Y = radius \* np.sin(angle)

plt.scatter(X, Y, c = angle, cmap = cm.hsv)

plt.show()

输出:



### 填充模式（fill pattern）

Code：

import numpy as np

import matplotlib.pyplot as plt

N = 8

A = np.random.random(N)

B = np.random.random(N)

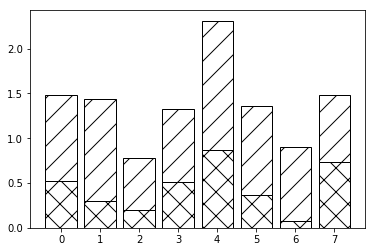
X = np.arange(N)

plt.bar(X, A, color = *'w'*, hatch = *'x'*, edgecolor=*'0.0'*)

plt.bar(X, A + B, bottom = A, color = *'w'*, hatch = *'/'*, edgecolor=*'0.0'*)

plt.show()

输出：



## 线条

### Style

*'-' solid line style*

*'--' dashed line style*

*'-.' dash-dot line style*

*':' dotted line style*

### 属性

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_agg\_filter', '\_alpha', '\_animated', '\_antialiased', '\_axes', '\_clipon', '\_clippath', '\_color', '\_contains', '\_dashOffset', '\_dashSeq', '\_dashcapstyle', '\_dashjoinstyle', '\_drawStyles\_l', '\_drawStyles\_s', '\_draw\_dash\_dot', '\_draw\_dashed', '\_draw\_dotted', '\_draw\_lines', '\_draw\_solid', '\_draw\_steps\_mid', '\_draw\_steps\_post', '\_draw\_steps\_pre', '\_drawstyle', '\_get\_markerfacecolor', '\_get\_rgba\_face', '\_get\_rgba\_ln\_color', '\_get\_transformed\_path', '\_gid', '\_invalidx', '\_invalidy', '\_is\_sorted', '\_label', '\_lineStyles', '\_linestyle', '\_linestyles', '\_linewidth', '\_marker', '\_markeredgecolor', '\_markeredgewidth', '\_markerfacecolor', '\_markerfacecoloralt', '\_markersize', '\_markevery', '\_mouseover', '\_oid', '\_path', '\_path\_effects', '\_picker', '\_prop\_order', '\_propobservers', '\_rasterized', '\_remove\_method', '\_set\_gc\_clip', '\_sketch', '\_snap', '\_solidcapstyle', '\_solidjoinstyle', '\_split\_drawstyle\_linestyle', '\_stale', '\_sticky\_edges', '\_subslice', '\_transform', '\_transformSet', '\_transform\_path', '\_transformed\_path', '\_url', '\_us\_dashOffset', '\_us\_dash

Seq', '\_visible', '\_x', '\_x\_filled', '\_xcid', '\_xorig', '\_xy', '\_y', '\_ycid', '\_yorig', 'add\_callback', 'aname', 'axes', 'clipbox', 'contains', 'convert\_xunits', 'convert\_yunits', 'draw', 'drawStyleKeys', 'drawStyles', 'eventson', 'figure', 'fillStyles', 'filled\_markers', 'findobj', 'format\_cursor\_data', 'get\_aa', 'get\_agg\_filter', 'get\_alpha', 'get\_animated', 'get\_antialiased', 'get\_axes', 'get\_c', 'get\_children', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_color', 'get\_contains', 'get\_cursor\_data', 'get\_dash\_capstyle', 'get\_dash\_joinstyle', 'get\_data', 'get\_drawstyle', 'get\_figure',

'get\_fillstyle', 'get\_gid', 'get\_label', 'get\_linestyle', 'get\_linewidth', 'get\_ls', 'get\_lw', 'get\_marker', 'get\_markeredgecolor', 'get\_markeredgewidth', 'get\_markerfacecolor', 'get\_markerfacecoloralt', 'get\_markersize', 'get\_markevery', 'get\_mec', 'get\_mew', 'get\_mfc', 'get\_mfcalt', 'get\_ms', 'get\_path', 'get\_path\_effects', 'get\_picker', 'get\_pickradius', 'get\_rasterized', 'get\_sketch\_params','get\_snap', 'get\_solid\_capstyle', 'get\_solid\_joinstyle', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_url', 'get\_visible', 'get\_window\_extent', 'get\_xdata', 'get\_xydata', 'get\_ydata',

'get\_zorder', 'have\_units', 'hitlist', 'ind\_offset', 'is\_dashed', 'is\_figure\_set', 'is\_transform\_set', 'lineStyles', 'markers', 'mouseover', 'pchanged', 'pick', 'pickable', 'pickradius', 'properties', 'recache', 'recache\_always', 'remove', 'remove\_callback', 'set', 'set\_aa', 'set\_agg\_filter', 'set\_alpha', 'set\_animated', 'set\_antialiased', 'set\_axes', 'set\_c', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_color', 'set\_contains', 'set\_dash\_capstyle', 'set\_dash\_joinstyle', 'set\_dashes', 'set\_data', 'set\_drawstyle', 'set\_figure', 'set\_fillstyle', 'set\_gid', 'set\_label', 'set\_linestyle', '

set\_linewidth', 'set\_ls', 'set\_lw', 'set\_marker', 'set\_markeredgecolor', 'set\_markeredgewidth', 'set\_markerfacecolor', 'set\_markerfacecoloralt', 'set\_markersize', 'set\_markevery', 'set\_mec', 'set\_mew', 'set\_mfc', 'set\_mfcalt', 'set\_ms', 'set\_path\_effects', 'set\_picker', 'set\_pickradius', 'set\_rasterized', 'set\_sketch\_params', 'set\_snap', 'set\_solid\_capstyle', 'set\_solid\_joinstyle', 'set\_transform', 'set\_url', 'set\_visible', 'set\_xdata', 'set\_ydata', 'set\_zorder', 'stale', 'stale\_callback', 'sticky\_edges', 'update', 'update\_from', 'validCap', 'validJoin', 'verticalOffset', 'zorder']

### 示例

Code:

import matplotlib.pyplot as plt

import numpy as np

def **show\_line\_styles**():

line\_style\_list=[*'-'*,*'--'*, *'-.'*, *':'*]

color\_list = [*'r'*, *'g'*,*'b'*,*'k'*]

for line, clr in zip(line\_style\_list,color\_list):

plt.plot(np.random.randn(10), linestyle=line, color=clr)

plt.show()

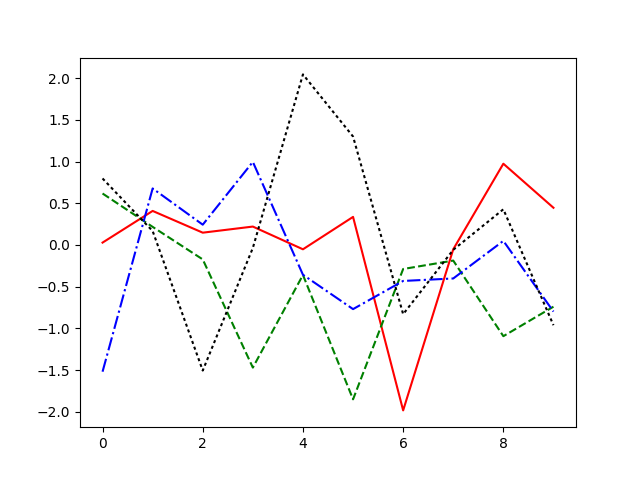
def **main**():

show\_line\_styles()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出:



## 标记 Marker

### type

*'.' 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*

### 属性

Type:<class 'matplotlib.collections.PathCollection'>:

Dir:

['\_A', '\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_agg\_filter', '\_alpha', '\_animated', '\_antialiaseds', '\_axes', '\_bcast\_lwls', '\_clipon', '\_clippath', '\_contains', '\_edge\_default', '\_edgecolors', '\_facecolors', '\_factor', '\_get\_bool', '\_get\_value', '\_gid', '\_hatch', '\_is\_filled', '\_is\_stroked', '\_label', '\_linestyles', '\_linewidths', '\_mouseover', '\_offset\_position', '\_offsets', '\_oid', '\_original\_edgecolor', '\_original\_facecolor', '\_path\_effects', '\_paths', '\_picker', '\_pickradius', '\_prepare\_points', '\_prop\_order', '\_propobservers', '\_rasterized', '\_remove\_method', '\_set\_edgecolor', '\_set\_facecolor', '\_set\_gc\_clip', '\_sizes', '\_sketch', '\_snap', '\_stale', '\_sticky\_edges', '\_transOffset', '\_transform', '\_transformSet', '\_transforms', '\_uniform\_offsets', '\_url', '\_urls', '\_us\_linestyles', '\_us\_lw', '\_visible', 'add\_callback', 'add\_checker', 'aname', 'autoscale', 'autoscale\_None', 'axes', 'callbacksSM', 'changed', 'check\_update', 'clipbox', 'cmap', 'colorbar', 'contains', 'convert\_xunits', 'convert\_yunits', 'draw', 'eventson', 'figure', 'findobj', 'format\_cursor\_data', 'get\_agg\_filter', 'get\_alpha', 'get\_animated', 'get\_array', 'get\_axes', 'get\_children', 'get\_clim', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_cmap', 'get\_contains', 'get\_cursor\_data', 'get\_dashes', 'get\_datalim', 'get\_edgecolor', 'get\_edgecolors', 'get\_facecolor', 'get\_facecolors', 'get\_figure', 'get\_fill', 'get\_gid', 'get\_hatch', 'get\_label', 'get\_linestyle', 'get\_linestyles', 'get\_linewidth', 'get\_linewidths', 'get\_offset\_position', 'get\_offset\_transform', 'get\_offsets', 'get\_path\_effects', 'get\_paths', 'get\_picker', 'get\_pickradius', 'get\_rasterized', 'get\_sizes', 'get\_sketch\_params', 'get\_snap', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_transforms', 'get\_url', 'get\_urls', 'get\_visible', 'get\_window\_extent', 'get\_zorder', 'have\_units', 'hitlist', 'is\_figure\_set', 'is\_transform\_set', 'mouseover', 'norm', 'pchanged', 'pick', 'pickable', 'properties', 'remove', 'remove\_callback', 'set', 'set\_agg\_filter', 'set\_alpha', 'set\_animated', 'set\_antialiased', 'set\_antialiaseds', 'set\_array', 'set\_axes', 'set\_clim', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_cmap', 'set\_color', 'set\_contains', 'set\_dashes', 'set\_edgecolor', 'set\_edgecolors', 'set\_facecolor', 'set\_facecolors', 'set\_figure', 'set\_gid', 'set\_hatch', 'set\_label', 'set\_linestyle', 'set\_linestyles', 'set\_linewidth', 'set\_linewidths', 'set\_lw', 'set\_norm', 'set\_offset\_position', 'set\_offsets', 'set\_path\_effects', 'set\_paths', 'set\_picker', 'set\_pickradius', 'set\_rasterized', 'set\_sizes', 'set\_sketch\_params', 'set\_snap', 'set\_transform', 'set\_url', 'set\_urls', 'set\_visible', 'set\_zorder', 'stale', 'stale\_callback', 'sticky\_edges', 'to\_rgba', 'update', 'update\_dict', 'update\_from', 'update\_scalarmappable', 'zorder']

### 示例

import matplotlib.pyplot as plt

import numpy as np

def **pickEvent**(event):

obj = event.artist

obj.set\_edgecolor(*'red'*)

obj.set\_alpha(1)

obj.set\_sizes([100])

plt.gcf().canvas.draw()

def **show\_marker\_styles**():

marker\_style\_list=[*'.'*, *','*, *'o'*, *'v'*, *'^'*, *'<'*, *'>'*, *'1'*, *'2'*, *'3'*, *'4'*, *'s'*, *'p'*, *'\*'*, *'h'*, *'H'*, *'+'*, *'x'*, *'D'*, *'d'*, *'|'*, *'\_'*]

xy\_ticks = np.arange(np.shape(marker\_style\_list)[0])

fig = plt.figure()

ax = fig.add\_subplot(111)

fig.canvas.mpl\_connect(*'pick\_event'*, pickEvent)

for mrk,xy\_tick in zip(marker\_style\_list, xy\_ticks):

sct = ax.scatter(xy\_tick, xy\_tick, marker=mrk, alpha=0.5, picker=5)

#sct.set\_picker(5)

plt.show()

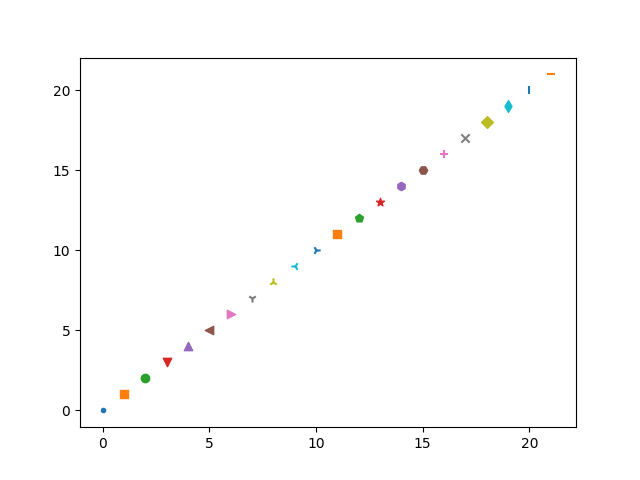
def **main**():

show\_marker\_styles()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出：



# 二维坐标平面

二维坐标平面的基本组成元素： Figure、Canvas和Axes

## Figure

### 创建Figure

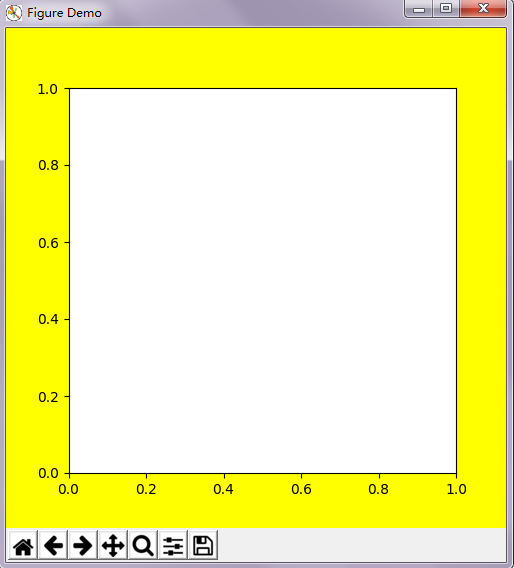
示例-1：

*import* matplotlib.pyplot *as* plt

fig = plt.figure(num='Figure Demo',figsize=(5,5), facecolor='yellow')  
ax = fig.add\_subplot(111)

plt.show()

输出一个大小为5x5英寸标题为“Figure Demo”的Figure：



### figure()函数原型

def **figure**(num=None, # autoincrement if None, else integer from 1-N

figsize=None, # defaults to rc figure.figsize

dpi=None, # defaults to rc figure.dpi

facecolor=None, # defaults to rc figure.facecolor

edgecolor=None, # defaults to rc figure.edgecolor

frameon=True,

FigureClass=Figure,

\*\*kwargs

)

查看figure函数说明：

print(plt.figure.\_\_doc\_\_)

输出：

Creates a new figure.

Parameters

----------

num : integer or string, optional, default: none

If not provided, a new figure will be created, and the figure number

will be incremented. The figure objects holds this number in a `number`

attribute.

If num is provided, and a figure with this id already exists, make

it active, and returns a reference to it. If this figure does not

exists, create it and returns it.

If num is a string, the window title will be set to this figure's

`num`.

figsize : tuple of integers, optional, default: None

width, height in inches. If not provided, defaults to rc

figure.figsize.

dpi : integer, optional, default: None

resolution of the figure. If not provided, defaults to rc figure.dpi.

facecolor :

the background color. If not provided, defaults to rc figure.facecolor

edgecolor :

the border color. If not provided, defaults to rc figure.edgecolor

**Returns**

-------

figure : Figure

The Figure instance returned will also be passed to new\_figure\_manager

in the backends, which allows to hook custom Figure classes into the pylab interface. Additional kwargs will be passed to the figure init function.

Notes

-----

If you are creating many figures, make sure you explicitly call "close"

on the figures you are not using, because this will enable pylab

to properly clean up the memory.

rcParams defines the default values, which can be modified in the matplotlibrc file

通过doc描述可知figure()函数返回的是一个Figure对象，通过type函数查看fig类型：

print(type(fig))

可以发现fig确实为一个 matplotlib.figure.Figure实例对象。

<class 'matplotlib.figure.Figure'>

查看fig属性：

print(dir(fig))

通过dir查看Figure的属性。

### Figure的属性：

>>>dir(fig)

['\_Figure\_\_remove\_ax', '\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_setstate\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_agg\_filter', '\_alpha', '\_animated', '\_axobservers', '\_axstack', '\_cachedRenderer', '\_clipon', '\_clippath', '\_contains', '\_dpi', '\_gci', '\_get\_axes', '\_get\_dpi', '\_gid', '\_hold', '\_label', '\_make\_key', '\_mouseover', '\_oid', '\_path\_effects', '\_picker', '\_prop\_order', '\_propobservers', '\_rasterized', '\_remove\_method', '\_repr\_html\_', '\_set\_artist\_props', '\_set\_dpi', '\_set\_gc\_clip', '\_sketch', '\_snap', '\_stale', '\_sticky\_edges', '\_suptitle', '\_tight', '\_tight\_parameters', '\_transform', '\_transformSet', '\_url', '\_visible', 'add\_axes', 'add\_axobserver', 'add\_callback', 'add\_subplot', 'aname', 'artists', 'autofmt\_xdate', 'axes', 'bbox', 'bbox\_inches', 'callbacks', 'canvas', 'clear', 'clf', 'clipbox', 'colorbar', 'contains', 'convert\_xunits', 'convert\_yunits', 'delaxes', 'dpi', 'dpi\_scale\_trans', 'draw', 'draw\_artist', 'eventson', 'figimage', 'figure', 'figurePatch', 'findobj', 'format\_cursor\_data', 'frameon', 'gca', 'get\_agg\_filter', 'get\_alpha', 'get\_animated', 'get\_axes', 'get\_children', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_contains', 'get\_cursor\_data', 'get\_default\_bbox\_extra\_artists', 'get\_dpi', 'get\_edgecolor', 'get\_facecolor', 'get\_figheight', 'get\_figure', 'get\_figwidth', 'get\_frameon', 'get\_gid', 'get\_label', 'get\_path\_effects', 'get\_picker', 'get\_rasterized', 'get\_size\_inches', 'get\_sketch\_params', 'get\_snap', 'get\_tight\_layout', 'get\_tightbbox', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_url', 'get\_visible', 'get\_window\_extent', 'get\_zorder', 'ginput', 'have\_units', 'hitlist', 'hold', 'images', 'is\_figure\_set', 'is\_transform\_set', 'legend', 'legends', 'lines', 'mouseover', 'number', 'patch', 'patches', 'pchanged', 'pick', 'pickable', 'properties', 'remove', 'remove\_callback', 'savefig', 'sca', 'set', 'set\_agg\_filter', 'set\_alpha', 'set\_animated', 'set\_axes', 'set\_canvas', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_contains', 'set\_dpi', 'set\_edgecolor', 'set\_facecolor', 'set\_figheight', 'set\_figure', 'set\_figwidth', 'set\_frameon', 'set\_gid', 'set\_label', 'set\_path\_effects', 'set\_picker', 'set\_rasterized', 'set\_size\_inches', 'set\_sketch\_params', 'set\_snap', 'set\_tight\_layout', 'set\_transform', 'set\_url', 'set\_visible', 'set\_zorder', 'show', 'stale', 'stale\_callback', 'sticky\_edges', 'subplotpars', 'subplots\_adjust', 'suppressComposite', 'suptitle', 'text', 'texts', 'tight\_layout', 'transFigure', 'update', 'update\_from', 'waitforbuttonpress', 'zorder']

### 客制化Figure

#### 增加子图

#### add\_subplot(\*args, \*\*kwargs)

#### Add a subplot. Examples:

fig.add\_subplot(111)

# equivalent but more general

fig.add\_subplot(1,1,1)

# add subplot with red background

fig.add\_subplot(212, facecolor='r')

# add a polar subplot

fig.add\_subplot(111, projection='polar')

# add Subplot instance sub

fig.add\_subplot(sub)

示例-2：

import matplotlib.pyplot as plt

fig = plt.figure()

ax1 = fig.add\_subplot(221)

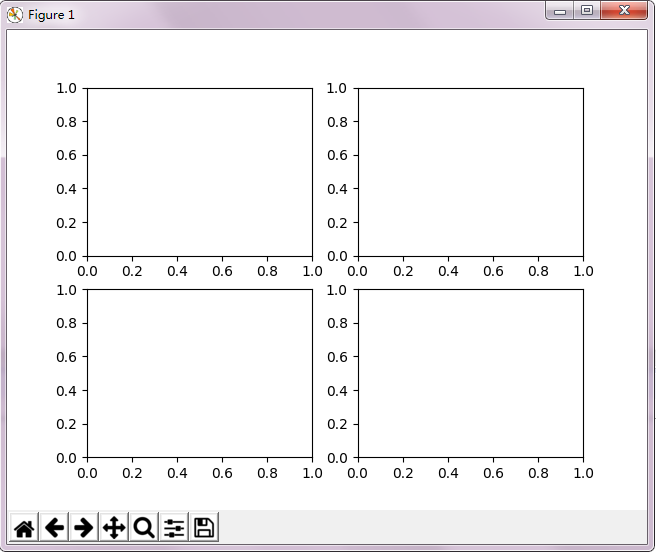
ax2 = fig.add\_subplot(222)

ax3 = fig.add\_subplot(223)

ax4 = fig.add\_subplot(224)

plt.show()

输出：



#### 增加子坐标

add\_axes(\*args, \*\*kwargs):

Add an axes at position rect [left, bottom, width, height] where all quantities are in fractions of figure width and height. kwargs are legal Axes kwargs plus projection which sets the projection type of the axes.

示例-3:

import matplotlib.pyplot as plt

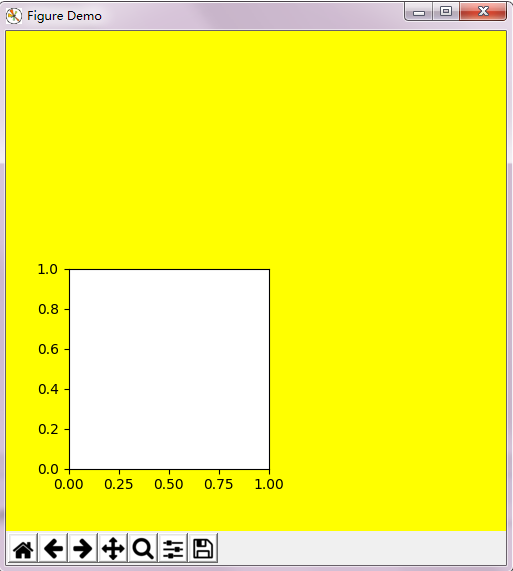
fig = plt.figure(num='Figure Demo',figsize=(5,5), facecolor='yellow',edgecolor='blue', frameon=True)

rect = 0.125,0.125,0.4,0.4

fig.add\_axes(rect, label='axes1')

plt.show()

输出:



## AxesSubplot

### 属性

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_setstate\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_add\_text', '\_adjustable', '\_agg\_filter', '\_alpha', '\_anchor', '\_animated', '\_aspect', '\_autoscaleXon', '\_autoscaleYon', '\_axes', '\_axes\_class', '\_axes\_locator', '\_axisbelow', '\_cachedRenderer', '\_clipon', '\_clippath', '\_connected', '\_contains', '\_current\_image', '\_cursorProps', '\_facecolor', '\_frameon', '\_gci', '\_gen\_axes\_patch', '\_gen\_axes\_spines', '\_get\_axis\_list', '\_get\_legend\_handles', '\_get\_lines', '\_get\_patches\_for\_fill', '\_get\_view', '\_gid', '\_gridOn', '\_hold', '\_init\_axis', '\_label', '\_left\_title', '\_make\_twin\_axes', '\_mouseover', '\_navigate', '\_navigate\_mode', '\_oid', '\_originalPosition', '\_path\_effects', '\_pcolorargs', '\_picker', '\_position', '\_process\_unit\_info', '\_prop\_order', '\_propobservers', '\_rasterization\_zorder', '\_rasterized', '\_remove\_method', '\_right\_title', '\_sci', '\_set\_artist\_props', '\_set\_gc\_clip', '\_set\_lim\_and\_transforms', '\_set\_view', '\_set\_view\_from\_bbox', '\_shared\_x\_axes', '\_shared\_y\_axes', '\_sharex', '\_sharey', '\_sketch', '\_snap', '\_stale', '\_sticky\_edges', '\_subplotspec', '\_tight', '\_transform', '\_transformSet', '\_update\_line\_limits', '\_update\_patch\_limits', '\_update\_transScale', '\_url', '\_use\_sticky\_edges', '\_visible', '\_xaxis\_transform', '\_xcid', '\_xmargin', '\_yaxis\_transform', '\_ycid', '\_ymargin', 'acorr', 'add\_artist', 'add\_callback', 'add\_collection', 'add\_container', 'add\_image', 'add\_line', 'add\_patch', 'add\_table', 'aname', 'angle\_spectrum', 'annotate', 'apply\_aspect', 'arrow', 'artists', 'autoscale', 'autoscale\_view', 'axes', 'axesPatch', 'axhline', 'axhspan', 'axis', 'axison', 'axvline', 'axvspan', 'bar', 'barbs', 'barh', 'bbox', 'boxplot', 'broken\_barh', 'bxp', 'callbacks', 'can\_pan', 'can\_zoom', 'change\_geometry', 'cla', 'clabel', 'clear', 'clipbox', 'cohere', 'colNum', 'collections', 'containers', 'contains', 'contains\_point', 'contour', 'contourf', 'convert\_xunits', 'convert\_yunits', 'csd', 'dataLim', 'drag\_pan', 'draw', 'draw\_artist', 'end\_pan', 'errorbar', 'eventplot', 'eventson', 'figbox', 'figure', 'fill', 'fill\_between', 'fill\_betweenx', 'findobj', 'fmt\_xdata', 'fmt\_ydata', 'format\_coord', 'format\_cursor\_data', 'format\_xdata', 'format\_ydata', 'get\_adjustable', 'get\_agg\_filter', 'get\_alpha', 'get\_anchor', 'get\_animated', 'get\_aspect', 'get\_autoscale\_on', 'get\_autoscalex\_on', 'get\_autoscaley\_on', 'get\_axes', 'get\_axes\_locator', 'get\_axis\_bgcolor', 'get\_axisbelow', 'get\_children', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_contains', 'get\_cursor\_data', 'get\_cursor\_props', 'get\_data\_ratio', 'get\_data\_ratio\_log', 'get\_default\_bbox\_extra\_artists', 'get\_facecolor', 'get\_fc', 'get\_figure', 'get\_frame\_on', 'get\_geometry', 'get\_gid', 'get\_images', 'get\_label', 'get\_legend', 'get\_legend\_handles\_labels', 'get\_lines', 'get\_navigate', 'get\_navigate\_mode', 'get\_path\_effects', 'get\_picker', 'get\_position', 'get\_rasterization\_zorder', 'get\_rasterized', 'get\_renderer\_cache', 'get\_shared\_x\_axes', 'get\_shared\_y\_axes', 'get\_sketch\_params', 'get\_snap', 'get\_subplotspec', 'get\_tightbbox', 'get\_title', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_url', 'get\_visible', 'get\_window\_extent', 'get\_xaxis', 'get\_xaxis\_text1\_transform', 'get\_xaxis\_text2\_transform', 'get\_xaxis\_transform', 'get\_xbound', 'get\_xgridlines', 'get\_xlabel', 'get\_xlim', 'get\_xmajorticklabels', 'get\_xminorticklabels', 'get\_xscale', 'get\_xticklabels', 'get\_xticklines', 'get\_xticks', 'get\_yaxis', 'get\_yaxis\_text1\_transform', 'get\_yaxis\_text2\_transform', 'get\_yaxis\_transform', 'get\_ybound', 'get\_ygridlines', 'get\_ylabel', 'get\_ylim', 'get\_ymajorticklabels', 'get\_yminorticklabels', 'get\_yscale', 'get\_yticklabels', 'get\_yticklines', 'get\_yticks', 'get\_zorder', 'grid', 'has\_data', 'have\_units', 'hexbin', 'hist', 'hist2d', 'hitlist', 'hlines', 'hold', 'ignore\_existing\_data\_limits', 'images', 'imshow', 'in\_axes', 'invert\_xaxis', 'invert\_yaxis', 'is\_figure\_set', 'is\_first\_col', 'is\_first\_row', 'is\_last\_col', 'is\_last\_row', 'is\_transform\_set', 'ishold', 'label\_outer', 'legend', 'legend\_', 'lines', 'locator\_params', 'loglog', 'magnitude\_spectrum', 'margins', 'matshow', 'minorticks\_off', 'minorticks\_on', 'mouseover', 'mouseover\_set', 'name', 'numCols', 'numRows', 'patch', 'patches', 'pchanged', 'pcolor', 'pcolorfast', 'pcolormesh', 'phase\_spectrum', 'pick', 'pickable', 'pie', 'plot', 'plot\_date', 'properties', 'psd', 'quiver', 'quiverkey', 'redraw\_in\_frame', 'relim', 'remove', 'remove\_callback', 'reset\_position', 'rowNum', 'scatter', 'semilogx', 'semilogy', 'set', 'set\_adjustable', 'set\_agg\_filter', 'set\_alpha', 'set\_anchor', 'set\_animated', 'set\_aspect', 'set\_autoscale\_on', 'set\_autoscalex\_on', 'set\_autoscaley\_on', 'set\_axes', 'set\_axes\_locator', 'set\_axis\_bgcolor', 'set\_axis\_off', 'set\_axis\_on', 'set\_axisbelow', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_color\_cycle', 'set\_contains', 'set\_cursor\_props', 'set\_facecolor', 'set\_fc', 'set\_figure', 'set\_frame\_on', 'set\_gid', 'set\_label', 'set\_navigate', 'set\_navigate\_mode', 'set\_path\_effects', 'set\_picker', 'set\_position', 'set\_prop\_cycle', 'set\_rasterization\_zorder', 'set\_rasterized', 'set\_sketch\_params', 'set\_snap', 'set\_subplotspec', 'set\_title', 'set\_transform', 'set\_url', 'set\_visible', 'set\_xbound', 'set\_xlabel', 'set\_xlim', 'set\_xmargin', 'set\_xscale', 'set\_xticklabels', 'set\_xticks', 'set\_ybound', 'set\_ylabel', 'set\_ylim', 'set\_ymargin', 'set\_yscale', 'set\_yticklabels', 'set\_yticks', 'set\_zorder', 'specgram', 'spines', 'spy', 'stackplot', 'stale', 'stale\_callback', 'start\_pan', 'stem', 'step', 'sticky\_edges', 'streamplot', 'table', 'tables', 'text', 'texts', 'tick\_params', 'ticklabel\_format', 'title', 'titleOffsetTrans', 'transAxes', 'transData', 'transLimits', 'transScale', 'tricontour', 'tricontourf', 'tripcolor', 'triplot', 'twinx', 'twiny', 'update', 'update\_datalim', 'update\_datalim\_bounds', 'update\_datalim\_numerix', 'update\_from', 'update\_params', 'use\_sticky\_edges', 'viewLim', 'violin', 'violinplot', 'vlines', 'xaxis', 'xaxis\_date', 'xaxis\_inverted', 'xcorr', 'yaxis', 'yaxis\_date', 'yaxis\_inverted', 'zorder']

### 示例

import matplotlib.pyplot as plt

def **figure\_demo**():

fig1 = plt.figure(num=*'This is figure-1'*, figsize=(5,5))

fig2 = plt.figure(num=*'This is figure-2'*, figsize=(5,5))

ax1\_1 = fig1.add\_subplot(211)

ax1\_2 = fig1.add\_subplot(212)

ax2\_1 = fig2.add\_subplot(211)

ax2\_2 = fig2.add\_subplot(212)

print(type(ax2\_2))

print(dir())

plt.show()

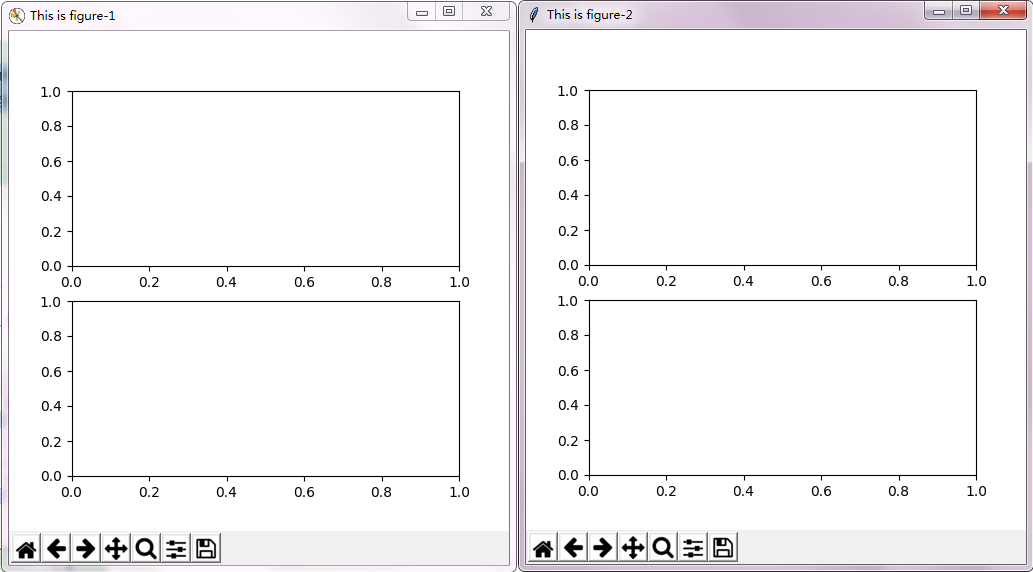
def **main**():

figure\_demo()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出：



## Axes

### 属性

Code：

import matplotlib.pyplot as plt

fig = plt.figure()

ax1 = fig.add\_subplot(111)

plt.show()

>>>type(ax1)

<class 'matplotlib.axes.\_subplots.AxesSubplot'>

通过dir查看ax1的属性：

>>>dir(ax1)

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_setstate\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_add\_text', '\_adjustable', '\_agg\_filter', '\_alpha', '\_anchor', '\_animated', '\_aspect', '\_autoscaleXon', '\_autoscaleYon', '\_axes', '\_axes\_class', '\_axes\_locator', '\_axisbelow', '\_cachedRenderer', '\_clipon', '\_clippath', '\_connected', '\_contains', '\_current\_image', '\_cursorProps', '\_facecolor', '\_frameon', '\_gci', '\_gen\_axes\_patch', '\_gen\_axes\_spines', '\_get\_axis\_list', '\_get\_legend\_handles', '\_get\_lines', '\_get\_patches\_for\_fill', '\_get\_view', '\_gid', '\_gridOn', '\_hold', '\_init\_axis', '\_label', '\_left\_title', '\_make\_twin\_axes', '\_mouseover', '\_navigate', '\_navigate\_mode', '\_oid', '\_originalPosition', '\_path\_effects', '\_pcolorargs', '\_picker', '\_position', '\_process\_unit\_info', '\_prop\_order', '\_propobservers', '\_rasterization\_zorder', '\_rasterized', '\_remove\_method', '\_right\_title', '\_sci', '\_set\_artist\_props', '\_set\_gc\_clip', '\_set\_lim\_and\_transforms', '\_set\_view', '\_set\_view\_from\_bbox', '\_shared\_x\_axes', '\_shared\_y\_axes', '\_sharex', '\_sharey', '\_sketch', '\_snap', '\_stale', '\_sticky\_edges', '\_subplotspec', '\_tight', '\_transform', '\_transformSet', '\_update\_line\_limits', '\_update\_patch\_limits', '\_update\_transScale', '\_url', '\_use\_sticky\_edges', '\_visible', '\_xaxis\_transform', '\_xcid', '\_xmargin', '\_yaxis\_transform', '\_ycid', '\_ymargin', 'acorr', 'add\_artist', 'add\_callback', 'add\_collection', 'add\_container', 'add\_image', 'add\_line', 'add\_patch', 'add\_table', 'aname', 'angle\_spectrum', 'annotate', 'apply\_aspect', 'arrow', 'artists', 'autoscale', 'autoscale\_view', 'axes', 'axesPatch', 'axhline', 'axhspan', 'axis', 'axison', 'axvline', 'axvspan', 'bar', 'barbs', 'barh', 'bbox', 'boxplot', 'broken\_barh', 'bxp', 'callbacks', 'can\_pan', 'can\_zoom', 'change\_geometry', 'cla', 'clabel', 'clear', 'clipbox', 'cohere', 'colNum', 'collections', 'containers', 'contains', 'contains\_point', 'contour', 'contourf', 'convert\_xunits', 'convert\_yunits', 'csd', 'dataLim', 'drag\_pan', 'draw', 'draw\_artist', 'end\_pan', 'errorbar', 'eventplot', 'eventson', 'figbox', 'figure', 'fill', 'fill\_between', 'fill\_betweenx', 'findobj', 'fmt\_xdata', 'fmt\_ydata', 'format\_coord', 'format\_cursor\_data', 'format\_xdata', 'format\_ydata', 'get\_adjustable', 'get\_agg\_filter', 'get\_alpha', 'get\_anchor', 'get\_animated', 'get\_aspect', 'get\_autoscale\_on', 'get\_autoscalex\_on', 'get\_autoscaley\_on', 'get\_axes', 'get\_axes\_locator', 'get\_axis\_bgcolor', 'get\_axisbelow', 'get\_children', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_contains', 'get\_cursor\_data', 'get\_cursor\_props', 'get\_data\_ratio', 'get\_data\_ratio\_log', 'get\_default\_bbox\_extra\_artists', 'get\_facecolor', 'get\_fc', 'get\_figure', 'get\_frame\_on', 'get\_geometry', 'get\_gid', 'get\_images', 'get\_label', 'get\_legend', 'get\_legend\_handles\_labels', 'get\_lines', 'get\_navigate', 'get\_navigate\_mode', 'get\_path\_effects', 'get\_picker', 'get\_position', 'get\_rasterization\_zorder', 'get\_rasterized', 'get\_renderer\_cache', 'get\_shared\_x\_axes', 'get\_shared\_y\_axes', 'get\_sketch\_params', 'get\_snap', 'get\_subplotspec', 'get\_tightbbox', 'get\_title', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_url', 'get\_visible', 'get\_window\_extent', 'get\_xaxis', 'get\_xaxis\_text1\_transform', 'get\_xaxis\_text2\_transform', 'get\_xaxis\_transform', 'get\_xbound', 'get\_xgridlines', 'get\_xlabel', 'get\_xlim', 'get\_xmajorticklabels', 'get\_xminorticklabels', 'get\_xscale', 'get\_xticklabels', 'get\_xticklines', 'get\_xticks', 'get\_yaxis', 'get\_yaxis\_text1\_transform', 'get\_yaxis\_text2\_transform', 'get\_yaxis\_transform', 'get\_ybound', 'get\_ygridlines', 'get\_ylabel', 'get\_ylim', 'get\_ymajorticklabels', 'get\_yminorticklabels', 'get\_yscale', 'get\_yticklabels', 'get\_yticklines', 'get\_yticks', 'get\_zorder', 'grid', 'has\_data', 'have\_units', 'hexbin', 'hist', 'hist2d', 'hitlist', 'hlines', 'hold', 'ignore\_existing\_data\_limits', 'images', 'imshow', 'in\_axes', 'invert\_xaxis', 'invert\_yaxis', 'is\_figure\_set', 'is\_first\_col', 'is\_first\_row', 'is\_last\_col', 'is\_last\_row', 'is\_transform\_set', 'ishold', 'label\_outer', 'legend', 'legend\_', 'lines', 'locator\_params', 'loglog', 'magnitude\_spectrum', 'margins', 'matshow', 'minorticks\_off', 'minorticks\_on', 'mouseover', 'mouseover\_set', 'name', 'numCols', 'numRows', 'patch', 'patches', 'pchanged', 'pcolor', 'pcolorfast', 'pcolormesh', 'phase\_spectrum', 'pick', 'pickable', 'pie', 'plot', 'plot\_date', 'properties', 'psd', 'quiver', 'quiverkey', 'redraw\_in\_frame', 'relim', 'remove', 'remove\_callback', 'reset\_position', 'rowNum', 'scatter', 'semilogx', 'semilogy', 'set', 'set\_adjustable', 'set\_agg\_filter', 'set\_alpha', 'set\_anchor', 'set\_animated', 'set\_aspect', 'set\_autoscale\_on', 'set\_autoscalex\_on', 'set\_autoscaley\_on', 'set\_axes', 'set\_axes\_locator', 'set\_axis\_bgcolor', 'set\_axis\_off', 'set\_axis\_on', 'set\_axisbelow', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_color\_cycle', 'set\_contains', 'set\_cursor\_props', 'set\_facecolor', 'set\_fc', 'set\_figure', 'set\_frame\_on', 'set\_gid', 'set\_label', 'set\_navigate', 'set\_navigate\_mode', 'set\_path\_effects', 'set\_picker', 'set\_position', 'set\_prop\_cycle', 'set\_rasterization\_zorder', 'set\_rasterized', 'set\_sketch\_params', 'set\_snap', 'set\_subplotspec', 'set\_title', 'set\_transform', 'set\_url', 'set\_visible', 'set\_xbound', 'set\_xlabel', 'set\_xlim', 'set\_xmargin', 'set\_xscale', 'set\_xticklabels', 'set\_xticks', 'set\_ybound', 'set\_ylabel', 'set\_ylim', 'set\_ymargin', 'set\_yscale', 'set\_yticklabels', 'set\_yticks', 'set\_zorder', 'specgram', 'spines', 'spy', 'stackplot', 'stale', 'stale\_callback', 'start\_pan', 'stem', 'step', 'sticky\_edges', 'streamplot', 'table', 'tables', 'text', 'texts', 'tick\_params', 'ticklabel\_format', 'title', 'titleOffsetTrans', 'transAxes', 'transData', 'transLimits', 'transScale', 'tricontour', 'tricontourf', 'tripcolor', 'triplot', 'twinx', 'twiny', 'update', 'update\_datalim', 'update\_datalim\_bounds', 'update\_datalim\_numerix', 'update\_from', 'update\_params', 'use\_sticky\_edges', 'viewLim', 'violin', 'violinplot', 'vlines', 'xaxis', 'xaxis\_date', 'xaxis\_inverted', 'xcorr', 'yaxis', 'yaxis\_date', 'yaxis\_inverted', 'zorder']

### Ticks

Code：

import matplotlib.pyplot as plt

import numpy as np

def **show\_ticks**():

fig = plt.figure()

ax = fig.add\_subplot(1,1,1)

ax.plot(np.arange(0,40,1),np.arange(0,1000,25))

xticks = ax.set\_xticks([0, 10, 20, 30, 40])

yticks = ax.set\_yticks([0, 250, 500, 750, 1000])

plt.show()

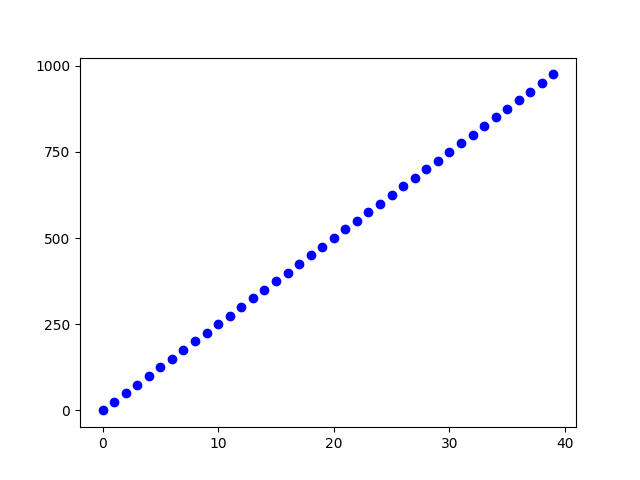
def **main**():

show\_ticks()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出：

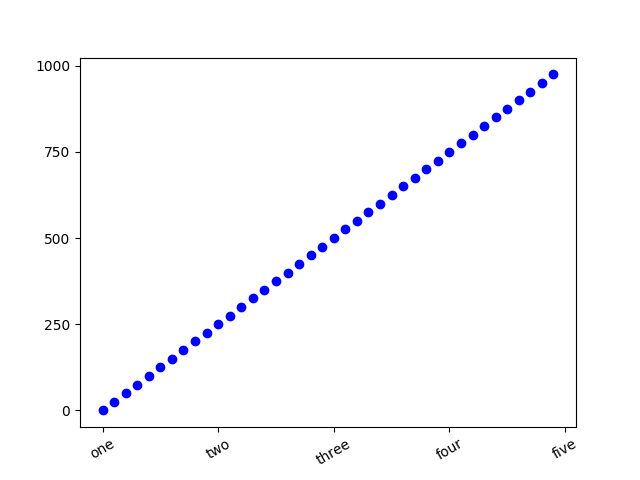


### Tick Labels

Code：

xlabels = ax.set\_xticklabels([*'one'*, *'two'*, *'three'*, *'four'*, *'five'*], rotation=30, fontsize=*'small'*)

输出：

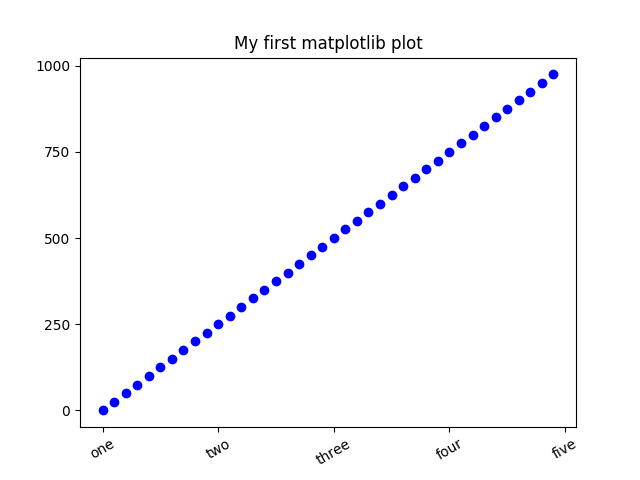


### Title

Code：

ax.set\_title('My first matplotlib plot')

输出：



### Labels

Code:

import matplotlib.pyplot as plt

def **show\_labels**():

fig = plt.figure()

ax = fig.add\_subplot(1,1,1)

ax.set\_xlabel(*'This is xlabel'* , color=*'green'*)

ax.set\_ylabel(*'This is ylabel'* , color=*'blue'*)

plt.show()

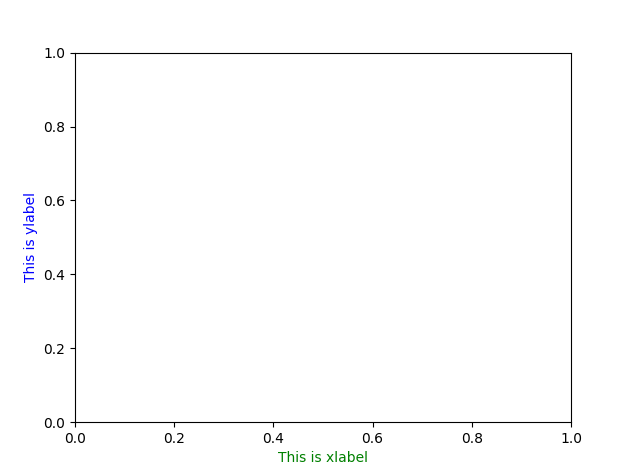
def **main**():

show\_labels()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出：



### Legends

#### 属性

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_agg\_filter', '\_alpha', '\_animated', '\_approx\_text\_height', '\_auto\_legend\_data', '\_axes', '\_bbox\_to\_anchor', '\_clipon', '\_clippath', '\_contains', '\_custom\_handler\_map', '\_default\_handler\_map', '\_draggable', '\_drawFrame', '\_find\_best\_position', '\_findoffset\_best', '\_findoffset\_loc', '\_fontsize', '\_get\_anchored\_bbox', '\_get\_loc', '\_gid', '\_init\_legend\_box', '\_label', '\_last\_fontsize\_points', '\_legend\_box', '\_legend\_handle\_box', '\_legend\_title\_box', '\_loc', '\_loc\_real', '\_mode', '\_mouseover', '\_ncol', '\_oid', '\_path\_effects', '\_picker', '\_prop\_order', '\_propobservers', '\_rasterized', '\_remove\_method', '\_scatteryoffsets', '\_set\_artist\_props', '\_set\_gc\_clip', '\_set\_loc', '\_sketch', '\_snap', '\_stale', '\_sticky\_edges', '\_transform', '\_transformSet', '\_url', '\_visible', 'add\_callback', 'aname', 'axes', 'borderaxespad', 'borderpad', 'clipbox', 'codes', 'columnspacing', 'contains', 'convert\_xunits', 'convert\_yunits', 'draggable', 'draw', 'draw\_frame', 'eventson', 'figure', 'findobj', 'format\_cursor\_data', 'get\_agg\_filter', 'get\_alpha', 'get\_animated', 'get\_axes', 'get\_bbox\_to\_anchor', 'get\_children', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_contains', 'get\_cursor\_data', 'get\_default\_handler\_map', 'get\_figure', 'get\_frame', 'get\_frame\_on', 'get\_gid', 'get\_label', 'get\_legend\_handler', 'get\_legend\_handler\_map', 'get\_lines', 'get\_patches', 'get\_path\_effects', 'get\_picker', 'get\_rasterized', 'get\_sketch\_params', 'get\_snap', 'get\_texts', 'get\_title', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_url', 'get\_visible', 'get\_window\_extent', 'get\_zorder', 'handleheight', 'handlelength', 'handletextpad', 'have\_units', 'hitlist', 'is\_figure\_set', 'is\_transform\_set', 'isaxes', 'labelspacing', 'legendHandles', 'legendPatch', 'markerscale', 'mouseover', 'numpoints', 'parent', 'pchanged', 'pick', 'pickable', 'prop', 'properties', 'remove', 'remove\_callback', 'scatterpoints', 'set', 'set\_agg\_filter', 'set\_alpha', 'set\_animated', 'set\_axes', 'set\_bbox\_to\_anchor', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_contains', 'set\_default\_handler\_map', 'set\_figure', 'set\_frame\_on', 'set\_gid', 'set\_label', 'set\_path\_effects', 'set\_picker', 'set\_rasterized', 'set\_sketch\_params', 'set\_snap', 'set\_title', 'set\_transform', 'set\_url', 'set\_visible', 'set\_zorder', 'shadow', 'stale', 'stale\_callback', 'sticky\_edges', 'texts', 'update', 'update\_default\_handler\_map', 'update\_from', 'zorder']

#### 示例

import numpy as np

import matplotlib.pyplot as plt

def **legend\_demo**():

fig = plt.figure()

ax1 = fig.add\_subplot(111)

x = np.linspace(0, np.pi\*2, 360)

y1 = np.sin(x)

y2 = np.cos(x)

ax1.plot(x,y1,*'r'*, label=*'sin(x)'*)

ax1.plot(x,y2,*'g'*, label=*'cos(x)'*)

ax1.legend(loc=*'best'*)

plt.show()

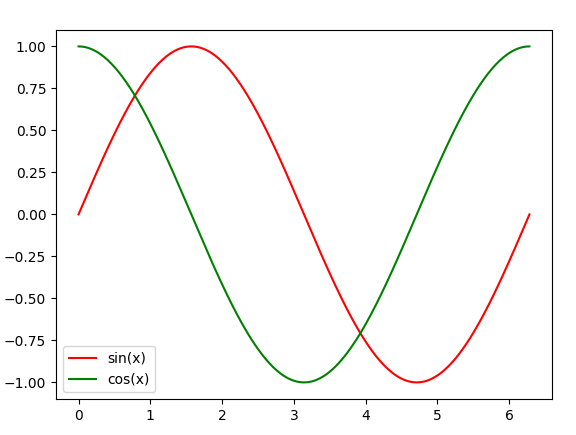
def **main**():

legend\_demo()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出：



### Text

Code:

import matplotlib.pyplot as plt

def **show\_text**():

fig = plt.figure()

ax = fig.add\_subplot(1,1,1)

ax.set\_xlim(0,10)

ax.set\_ylim(0,10)

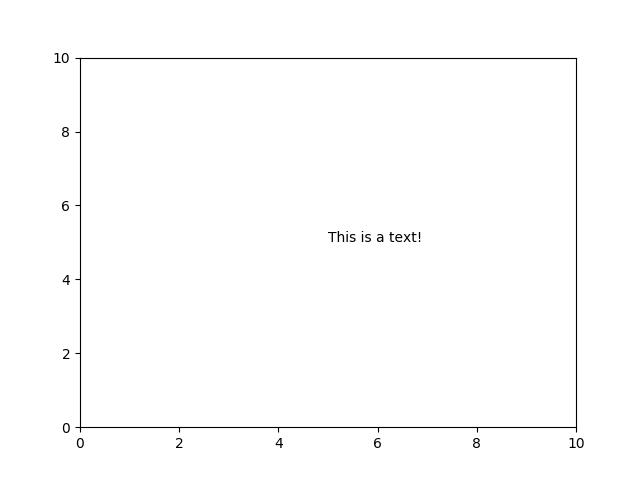
ax.text(5,5, *"This is a text!"*)

plt.show()

if \_\_name\_\_ == *'\_\_main\_\_'*:

show\_text()

输出：



### Color

Code：

import matplotlib.pyplot as plt

def **show\_text**():

fig = plt.figure()

ax = fig.add\_subplot(1,1,1)

ax.set\_xlim(0,10)

ax.set\_ylim(0,10)

#set text color

ax.text(5,5, *"This is a text!"*, color=*'green'*)

#set background color

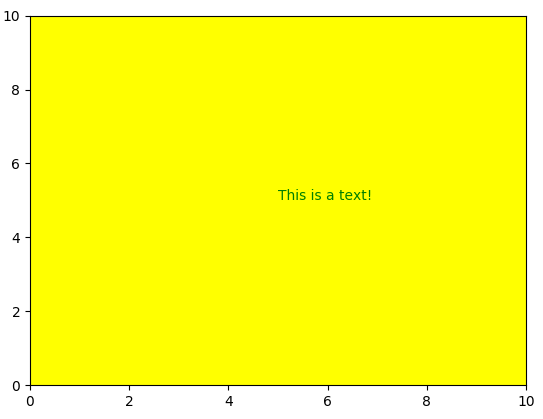
ax.set\_facecolor(*'yellow'*)

plt.show()

if \_\_name\_\_ == *'\_\_main\_\_'*:

show\_text()

输出：



### 坐标纵横比(Aspect Ratio)

以下通过画一个圆展示坐标纵横比例的使用。

Code：

import matplotlib.pyplot as plt

import numpy as np

def **draw\_circle**(ax, r=1):

theta = np.linspace(start=0, stop=2\*np.pi, num=360, endpoint=True)

y = r\*np.sin(theta)

x = r\*np.cos(theta)

ax.plot(x,y)

def **aspectRatio\_demo**():

fig = plt.figure()

ax = fig.add\_subplot(111)

draw\_circle(ax, r=2)

plt.show()

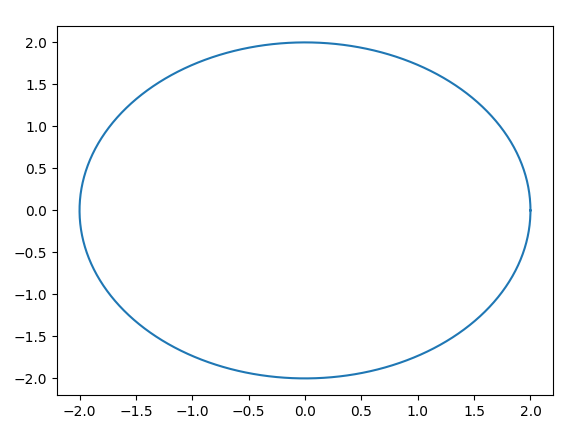
def **main**():

aspectRatio\_demo()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出：



从输出结果来看，得到的是一个椭圆，而不是圆。导致这种结果的原因是，x-y轴的坐标单位不一致。

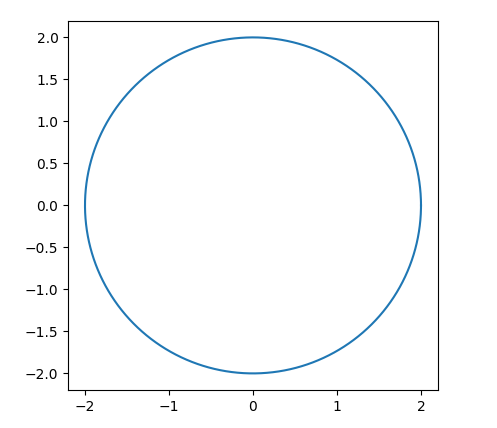
要统一x-y轴的坐标单位，可以通过以下语句实现：

ax.set\_aspect(1)

或者：

ax.set\_aspect(*'equal'*)

再看输出：



这时输出就正常了。

## Spines

### 属性

Code:

import matplotlib.pyplot as plt

ax = plt.gca()

spines = ax.spines

通过type命令查看一下spines的类型：

>>>type(spines)

<class 'collections.OrderedDict'>

由此可知为OrderedDict字典类型，那么我们可以通过访问字典的方式查看它有哪些key值：

>>>spines.keys()

odict\_keys(['left', 'right', 'bottom', 'top'])

spines包含左、右、上、下四部分，可以分别通过spines['left']、spines['right']、spines['bottom']、spines['top']来访问。查看一下spines['left']的类型：

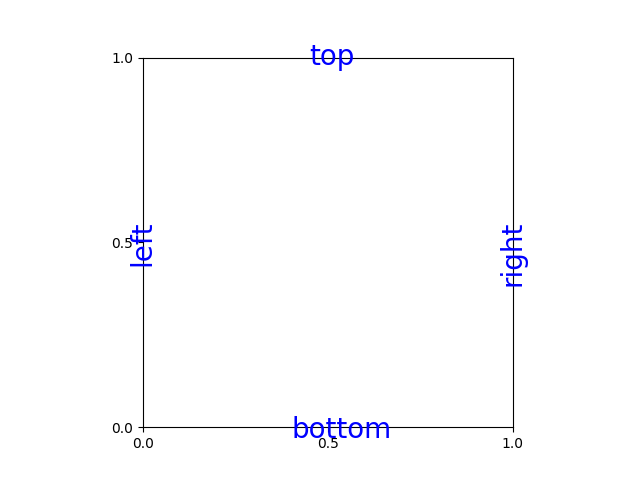
>>> type(spines['left'])

<class 'matplotlib.spines.Spine'>

dir查看matplotlib.spines.Spine包含哪些属性：

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_adjust\_location', '\_agg\_filter', '\_alpha', '\_animated', '\_antialiased', '\_axes', '\_bounds', '\_calc\_offset\_transform', '\_capstyle', '\_clipon', '\_clippath', '\_combined\_transform', '\_contains', '\_dashes', '\_dashoffset', '\_edge\_default', '\_edgecolor', '\_ensure\_position\_is\_set', '\_facecolor', '\_fill', '\_gid', '\_hatch', '\_joinstyle', '\_label', '\_linestyle', '\_linewidth', '\_mouseover', '\_oid', '\_original\_edgecolor', '\_original\_facecolor', '\_patch\_transform', '\_patch\_type', '\_path', '\_path\_effects', '\_picker', '\_position', '\_process\_radius', '\_prop\_order', '\_propobservers', '\_rasterized', '\_recompute\_transform', '\_remove\_method', '\_set\_edgecolor', '\_set\_facecolor', '\_set\_gc\_clip', '\_sketch', '\_smart\_bounds', '\_snap', '\_spine\_transform', '\_stale', '\_sticky\_edges', '\_transform', '\_transformSet', '\_url', '\_us\_dashes', '\_visible', 'add\_callback', 'aname', 'axes', 'axis', 'circular\_spine', 'cla', 'clipbox', 'contains', 'contains\_point', 'convert\_xunits', 'convert\_yunits', 'draw', 'eventson', 'figure', 'fill', 'findobj', 'format\_cursor\_data', 'get\_aa', 'get\_agg\_filter', 'get\_alpha', 'get\_animated', 'get\_antialiased', 'get\_axes', 'get\_bounds', 'get\_capstyle', 'get\_children', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_contains', 'get\_cursor\_data', 'get\_data\_transform', 'get\_ec', 'get\_edgecolor', 'get\_extents', 'get\_facecolor', 'get\_fc', 'get\_figure', 'get\_fill', 'get\_gid', 'get\_hatch', 'get\_joinstyle', 'get\_label', 'get\_linestyle', 'get\_linewidth', 'get\_ls', 'get\_lw', 'get\_patch\_transform', 'get\_path', 'get\_path\_effects', 'get\_picker', 'get\_position', 'get\_rasterized', 'get\_sketch\_params', 'get\_smart\_bounds', 'get\_snap', 'get\_spine\_transform', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_url', 'get\_verts', 'get\_visible', 'get\_window\_extent', 'get\_zorder', 'have\_units', 'hitlist', 'is\_figure\_set', 'is\_frame\_like', 'is\_transform\_set', 'linear\_spine', 'mouseover', 'pchanged', 'pick', 'pickable', 'properties', 'register\_axis', 'remove', 'remove\_callback', 'set', 'set\_aa', 'set\_agg\_filter', 'set\_alpha', 'set\_animated', 'set\_antialiased', 'set\_axes', 'set\_bounds', 'set\_capstyle', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_color', 'set\_contains', 'set\_ec', 'set\_edgecolor', 'set\_facecolor', 'set\_fc', 'set\_figure', 'set\_fill', 'set\_gid', 'set\_hatch', 'set\_joinstyle', 'set\_label', 'set\_linestyle', 'set\_linewidth', 'set\_ls', 'set\_lw', 'set\_patch\_circle', 'set\_patch\_line', 'set\_path\_effects', 'set\_picker', 'set\_position', 'set\_rasterized', 'set\_sketch\_params', 'set\_smart\_bounds', 'set\_snap', 'set\_transform', 'set\_url', 'set\_visible', 'set\_zorder', 'spine\_type', 'stale', 'stale\_callback', 'sticky\_edges', 'update', 'update\_from', 'validCap', 'validJoin', 'zorder']

### 坐标定制



坐标轴刺

坐标轴刺由上、下、左、右4部分组成。

### 坐标轴刺的显示控制

隐藏top、right轴刺：

Code：

import matplotlib.pyplot as plt

def **hide\_axes**():

ax = plt.gca()

ax.spines[*'top'*]. set\_visible(False)

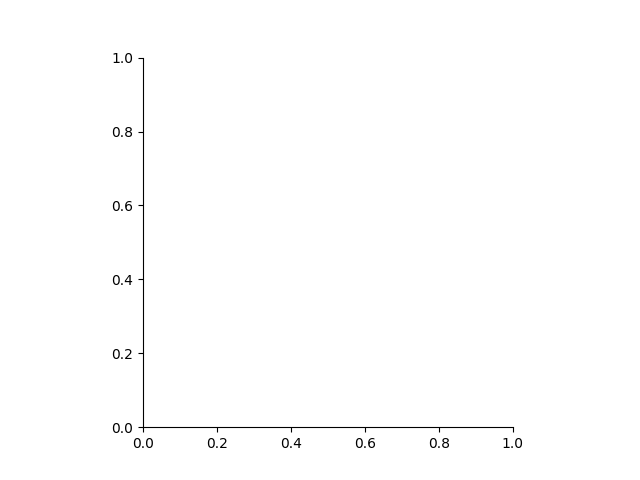
ax.spines[*'right'*].set\_color(*'none'*)

plt.show()

if \_\_name\_\_ == *'\_\_main\_\_'*:

hide\_axes()

输出：



### 定制坐标位置

Code：

import matplotlib.pyplot as plt

def **customize\_axes**():

ax = plt.gca()

ax.set\_xlim(-8,8)

ax.set\_ylim(-4,4)

#hide top & right spines

ax.spines[*'right'*].set\_visible(False)

ax.spines[*'top'*].set\_visible(False)

#set the position of the spines

ax.spines[*'bottom'*].set\_position(*'zero'*) # or 'center'

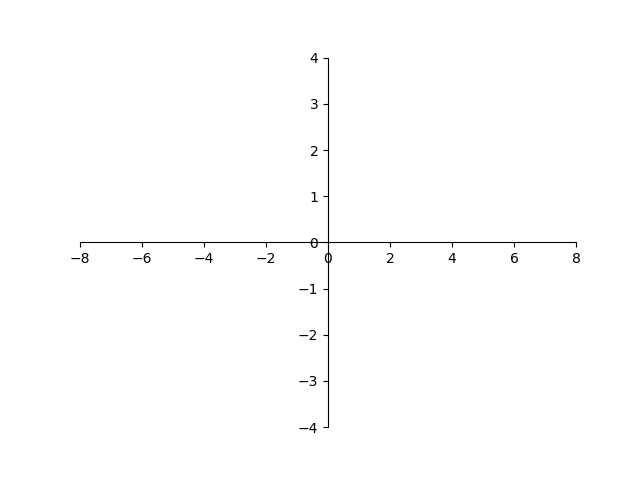
ax.spines[*'left'*].set\_position(*'zero'*)

plt.show()

if \_\_name\_\_ == *'\_\_main\_\_'*:

customize\_axes()

输出：



## Canvas

matplotlib所有的图形绘制的操作都是在canvas上完成的。

### 属性

Code:

import matplotlib.pyplot as plt

fig = plt.figure()

fig\_canvas = fig.canvas

查看fig\_canvas类型：

>>>type(fig\_canvas)

<class 'matplotlib.backends.backend\_tkagg.FigureCanvasTkAgg'>

从这里看出fig\_canvas的类型为：

matplotlib.backends.backend\_tkagg.FigureCanvasTkAgg

但实际这与matplotlib的backend选择是有关的。我们查看一下matplolib当前使用的的backend:

>>> plt.get\_backend()

'TkAgg'

可以通过matplotlib.use()函数来更改matplotlib的backend,其合法的backend有：

['GTK3Cairo', 'WXAgg', 'svg', 'GTK', 'TkAgg', 'agg', 'nbAgg', 'pgf', 'MacOSX', 'GTKAgg', 'GTKCairo', 'WX', 'Qt4Agg', 'pdf', 'Qt5Agg', 'ps', 'GTK3Agg', 'cairo', 'WebAgg', 'gdk', 'template']

下面我们将backend改成Qt5:

import matplotlib

matplotlib.use(*'Qt5Agg'*)

import matplotlib.pyplot as plt

fig = plt.figure()

print(plt.get\_backend())

这里需要注意的是，matplotlib.use(*'**Qt5Agg'*)语句必须在import matplotlib语句后import matplotlib.pyplot as plt语句前。

此时plt的backend已经变成了*Qt5Agg*了。

>>>type(fig\_canvas)

<class 'matplotlib.backends.backend\_qt5agg.FigureCanvasQTAgg'>

对于不同的backend拥有的属性也不同，如果backend是*TkAgg*对应属性为：

>>>dir(fig\_canvas)

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_button', '\_get\_key', '\_get\_output\_canvas', '\_idle', '\_idle\_callback', '\_idle\_draw\_cntx', '\_is\_idle\_drawing', '\_is\_saving', '\_key', '\_keycode\_lookup', '\_lastx', '\_lasty', '\_master', '\_resize\_callback', '\_tkcanvas', '\_tkphoto', '\_update\_pointer\_position', 'blit', 'buffer\_rgba', 'button\_dblclick\_event', 'button\_pick\_id', 'button\_press\_event', 'button\_release\_event', 'callbacks', 'close\_event', 'copy\_from\_bbox', 'draw', 'draw\_cursor', 'draw\_event', 'draw\_idle', 'enter\_notify\_event', 'events', 'figure', 'filetypes', 'fixed\_dpi', 'flush\_events', 'get\_default\_filename', 'get\_default\_filetype', 'get\_renderer', 'get\_supported\_filetypes', 'get\_supported\_filetypes\_grouped', 'get\_tk\_widget', 'get\_width\_height', 'get\_window\_title', 'grab\_mouse', 'idle\_event', 'is\_saving', 'key\_press', 'key\_press\_event', 'key\_release', 'key\_release\_event', 'keyvald', 'leave\_notify\_event', 'manager', 'motion\_notify\_event', 'mouse\_grabber', 'mpl\_connect', 'mpl\_disconnect', 'new\_timer', 'onHilite', 'onRemove', 'pick', 'pick\_event', 'print\_figure', 'print\_jpeg', 'print\_jpg', 'print\_png', 'print\_raw', 'print\_rgba', 'print\_tif', 'print\_tiff', 'print\_to\_buffer', 'release\_mouse', 'resize', 'resize\_event', 'restore\_region', 'scroll\_event', 'scroll\_event\_windows', 'scroll\_pick\_id', 'set\_window\_title', 'show', 'start\_event\_loop', 'start\_event\_loop\_default', 'stop\_event\_loop', 'stop\_event\_loop\_default', 'supports\_blit', 'switch\_backends', 'toolbar', 'tostring\_argb', 'tostring\_rgb', 'widgetlock']

如果backend为*Qt5Agg*，则其属性为：

['DrawChildren', 'DrawWindowBackground', 'IgnoreMask', 'PaintDeviceMetric', 'PdmDepth', 'PdmDevicePixelRatio', 'PdmDevicePixelRatioScaled', 'PdmDpiX', 'PdmDpiY', 'PdmHeight', 'PdmHeightMM', 'PdmNumColors', 'PdmPhysicalDpiX', 'PdmPhysicalDpiY', 'PdmWidth', 'PdmWidthMM', 'RenderFlag', 'RenderFlags', '\_FigureCanvasQTAggBase\_\_draw\_idle\_agg', '\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattr\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_agg\_draw\_pending', '\_button', '\_dpi\_ratio', '\_drawRect', '\_get\_key', '\_get\_output\_canvas', '\_idle\_draw\_cntx', '\_is\_idle\_drawing', '\_is\_saving', '\_key', '\_lastx', '\_lasty', 'acceptDrops', 'accessibleDescription', 'accessibleName', 'actionEvent', 'actions', 'activateWindow', 'addAction', 'addActions', 'adjustSize', 'autoFillBackground', 'backgroundRole', 'baseSize', 'blit', 'blitbox', 'blockSignals', 'buffer\_rgba', 'button\_pick\_id', 'button\_press\_event', 'button\_release\_event', 'buttond', 'callbacks', 'changeEvent', 'childAt', 'childEvent', 'children', 'childrenRect', 'childrenRegion', 'clearFocus', 'clearMask', 'close', 'closeEvent', 'close\_event', 'colorCount', 'connectNotify', 'contentsMargins', 'contentsRect', 'contextMenuEvent', 'contextMenuPolicy', 'copy\_from\_bbox', 'create', 'createWindowContainer', 'cursor', 'customContextMenuRequested', 'customEvent', 'deleteLater', 'depth', 'destroy', 'destroyed', 'devType', 'devicePixelRatio', 'devicePixelRatioF', 'devicePixelRatioFScale', 'disconnect', 'disconnectNotify', 'dragEnterEvent', 'dragLeaveEvent', 'dragMoveEvent', 'draw', 'drawRectangle', 'draw\_cursor', 'draw\_event', 'draw\_idle', 'dropEvent', 'dumpObjectInfo', 'dumpObjectTree', 'dynamicPropertyNames', 'effectiveWinId', 'ensurePolished', 'enterEvent', 'enter\_notify\_event', 'event', 'eventFilter', 'events', 'figure', 'filetypes', 'find', 'findChild', 'findChildren', 'fixed\_dpi', 'flush\_events', 'focusInEvent', 'focusNextChild', 'focusNextPrevChild', 'focusOutEvent', 'focusPolicy', 'focusPreviousChild', 'focusProxy', 'focusWidget', 'font', 'fontInfo', 'fontMetrics', 'foregroundRole', 'frameGeometry', 'frameSize', 'geometry', 'getContentsMargins', 'get\_default\_filename', 'get\_default\_filetype', 'get\_renderer', 'get\_supported\_filetypes', 'get\_supported\_filetypes\_grouped', 'get\_width\_height', 'get\_window\_title', 'grab', 'grabGesture', 'grabKeyboard', 'grabMouse', 'grabShortcut', 'grab\_mouse', 'graphicsEffect', 'graphicsProxyWidget', 'hasFocus', 'hasHeightForWidth', 'hasMouseTracking', 'height', 'heightForWidth', 'heightMM', 'hide', 'hideEvent', 'idle\_event', 'inherits', 'initPainter', 'inputMethodEvent', 'inputMethodHints', 'inputMethodQuery', 'insertAction', 'insertActions', 'installEventFilter', 'isActiveWindow', 'isAncestorOf', 'isEnabled', 'isEnabledTo', 'isFullScreen', 'isHidden', 'isLeftToRight', 'isMaximized', 'isMinimized', 'isModal', 'isRightToLeft', 'isSignalConnected', 'isVisible', 'isVisibleTo', 'isWidgetType', 'isWindow', 'isWindowModified', 'isWindowType', 'is\_saving', 'keyPressEvent', 'keyReleaseEvent', 'key\_press\_event', 'key\_release\_event', 'keyboardGrabber', 'killTimer', 'layout', 'layoutDirection', 'leaveEvent', 'leave\_notify\_event', 'locale', 'logicalDpiX', 'logicalDpiY', 'lower', 'manager', 'mapFrom', 'mapFromGlobal', 'mapFromParent', 'mapTo', 'mapToGlobal', 'mapToParent', 'mask', 'maximumHeight', 'maximumSize', 'maximumWidth', 'metaObject', 'metric', 'minimumHeight', 'minimumSize', 'minimumSizeHint', 'minimumWidth', 'minumumSizeHint', 'motion\_notify\_event', 'mouseDoubleClickEvent', 'mouseEventCoords', 'mouseGrabber', 'mouseMoveEvent', 'mousePressEvent', 'mouseReleaseEvent', 'mouse\_grabber', 'move', 'moveEvent', 'moveToThread', 'mpl\_connect', 'mpl\_disconnect', 'nativeEvent', 'nativeParentWidget', 'new\_timer', 'nextInFocusChain', 'normalGeometry', 'objectName', 'objectNameChanged', 'onHilite', 'onRemove', 'overrideWindowFlags', 'overrideWindowState', 'paintEngine', 'paintEvent', 'paintingActive', 'palette', 'parent', 'parentWidget', 'physicalDpiX', 'physicalDpiY', 'pick', 'pick\_event', 'pos', 'previousInFocusChain', 'print\_figure', 'print\_jpeg', 'print\_jpg', 'print\_png', 'print\_raw', 'print\_rgba', 'print\_tif', 'print\_tiff', 'print\_to\_buffer', 'property', 'pyqtConfigure', 'raise\_', 'receivers', 'rect', 'releaseKeyboard', 'releaseMouse', 'releaseShortcut', 'release\_mouse', 'removeAction', 'removeEventFilter', 'render', 'repaint', 'resize', 'resizeEvent', 'resize\_event', 'restoreGeometry', 'restore\_region', 'saveGeometry', 'scroll', 'scroll\_event', 'scroll\_pick\_id', 'sender', 'senderSignalIndex', 'setAcceptDrops', 'setAccessibleDescription', 'setAccessibleName', 'setAttribute', 'setAutoFillBackground', 'setBackgroundRole', 'setBaseSize', 'setContentsMargins', 'setContextMenuPolicy', 'setCursor', 'setDisabled', 'setEnabled', 'setFixedHeight', 'setFixedSize', 'setFixedWidth', 'setFocus', 'setFocusPolicy', 'setFocusProxy', 'setFont', 'setForegroundRole', 'setGeometry', 'setGraphicsEffect', 'setHidden', 'setInputMethodHints', 'setLayout', 'setLayoutDirection', 'setLocale', 'setMask', 'setMaximumHeight', 'setMaximumSize', 'setMaximumWidth', 'setMinimumHeight', 'setMinimumSize', 'setMinimumWidth', 'setMouseTracking', 'setObjectName', 'setPalette', 'setParent', 'setProperty', 'setShortcutAutoRepeat', 'setShortcutEnabled', 'setSizeIncrement', 'setSizePolicy', 'setStatusTip', 'setStyle', 'setStyleSheet', 'setTabOrder', 'setToolTip', 'setToolTipDuration', 'setUpdatesEnabled', 'setVisible', 'setWhatsThis', 'setWindowFilePath', 'setWindowFlags', 'setWindowIcon', 'setWindowIconText', 'setWindowModality', 'setWindowModified', 'setWindowOpacity', 'setWindowRole', 'setWindowState', 'setWindowTitle', 'set\_window\_title', 'sharedPainter', 'show', 'showEvent', 'showFullScreen', 'showMaximized', 'showMinimized', 'showNormal', 'signalsBlocked', 'size', 'sizeHint', 'sizeIncrement', 'sizePolicy', 'stackUnder', 'startTimer', 'start\_event\_loop', 'start\_event\_loop\_default', 'staticMetaObject', 'statusTip', 'stop\_event\_loop', 'stop\_event\_loop\_default', 'style', 'styleSheet', 'supports\_blit', 'switch\_backends', 'tabletEvent', 'testAttribute', 'thread', 'timerEvent', 'toolTip', 'toolTipDuration', 'toolbar', 'tostring\_argb', 'tostring\_rgb', 'tr', 'underMouse', 'ungrabGesture', 'unsetCursor', 'unsetLayoutDirection', 'unsetLocale', 'update', 'updateGeometry', 'updateMicroFocus', 'updatesEnabled', 'visibleRegion', 'whatsThis', 'wheelEvent', 'widgetlock', 'width', 'widthMM', 'winId', 'window', 'windowFilePath', 'windowFlags', 'windowHandle', 'windowIcon', 'windowIconChanged', 'windowIconText', 'windowIconTextChanged', 'windowModality', 'windowOpacity', 'windowRole', 'windowState', 'windowTitle', 'windowTitleChanged', 'windowType', 'x', 'y']

# 应用交互

## 事件处理

### 按键处理

Code:

import matplotlib.pyplot as plt

def **process\_key\_press**(event):

print(*"Key pressed:"*, event.key)

def **process\_key\_rls**(event):

print(*"Key released:"*, event.key)

fig, ax = plt.subplots(1, 1)

kprs\_id = fig.canvas.mpl\_connect(*'key\_press\_event'*, process\_key\_press)

krls\_id = fig.canvas.mpl\_connect(*'key\_release\_event'*, process\_key\_rls)

输出：

Key pressed: a

Key released: a

Key pressed: b

Key released: b

Key pressed: c

Key released: c

代码分析：

Debug进入到process\_key\_press函数对event变量进行分析：

>>> type(event)

<class 'matplotlib.backend\_bases.KeyEvent'>

由此我们知道event变量的类型为matplotlib.backend\_bases.KeyEvent，继续通过dir查看event的属性：

>>> dir(event)

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_update\_enter\_leave', 'canvas', 'guiEvent', 'inaxes', 'key', 'lastevent', 'name', 'x', 'xdata', 'y', 'ydata']

对key event有意义的几个属性：

key: 键值

inaxes: key事件发生所在的坐标实例

### 取消按键响应

fig.canvas.mpl\_disconnect(kprs\_id)

fig.canvas.mpl\_disconnect(krls\_id)

### 鼠标事件处理

#### 鼠标点击事件

button\_press\_event 、button\_release\_event

Code:

import matplotlib.pyplot as plt

def **process\_button\_press**(event):

print(*"Button:"*, event.x, event.y, event.xdata, event.ydata,event.button)

fig, ax = plt.subplots(1, 1)

fig.canvas.mpl\_connect(*'button\_press\_event'*, process\_button\_press)

plt.show()

输出

鼠标左键：

Button: 141 171.0 0.122983870968 0.319805194805 1

鼠标中键：

Button: 141 171.0 0.122983870968 0.319805194805 2

鼠标右键：

Button: 141 171.0 0.122983870968 0.319805194805 3

代码分析：

Debug进入到process\_button\_press函数对event变量进行分析：

>>> type(event)

<class 'matplotlib.backend\_bases.MouseEvent'>

dir查看event变量的属性：

>>> dir(event)

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_update\_enter\_leave', 'button', 'canvas', 'dblclick', 'guiEvent', 'inaxes', 'key', 'lastevent', 'name', 'step', 'x', 'xdata', 'y', 'ydata']

单击鼠标时：

>>> event.dblclick

False

双击鼠标时：

双击第一次event.dblclick为False;双击第二次event.dblclick为True;如果连续3次或3次以上点击鼠标，第一次event.dblclick为False,后面每次event.dblclick均为True。

#### 鼠标移动事件

*motion\_notify\_event*

Code：

import matplotlib.pyplot as plt

def **process\_motion\_notify**(event):

print(*"mouse:"*, event.x, event.y, event.xdata, event.ydata)

fig, ax = plt.subplots(1, 1)

fig.canvas.mpl\_connect(*'motion\_notify\_event'*, process\_motion\_notify)

plt.show()

输出：

mouse: 451 167.0 0.747983870968 0.308982683983

mouse: 449 166.0 0.743951612903 0.306277056277

mouse: 449 163.0 0.743951612903 0.29816017316

mouse: 448 163.0 0.741935483871 0.29816017316

mouse: 448 162.0 0.741935483871 0.295454545455

mouse: 446 161.0 0.737903225806 0.292748917749

mouse: 446 160.0 0.737903225806 0.290043290043

代码分析：

Debug进入到process\_button\_press函数对event变量进行分析：

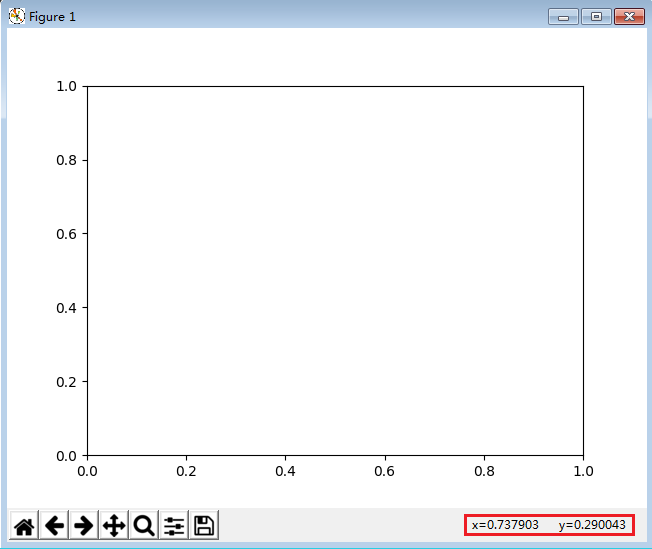
>>> type(event)

<class 'matplotlib.backend\_bases.MouseEvent'>

>>> dir(event)

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_update\_enter\_leave', 'button', 'canvas', 'dblclick', 'guiEvent', 'inaxes', 'key', 'lastevent', 'name', 'step', 'x', 'xdata', 'y', 'ydata']

观察Figure右下角的x,y值，发现xdata,ydata其实就是鼠标所在坐标轴的x、y轴的坐标值。



#### 鼠标滚动事件

Code:

import matplotlib.pyplot as plt

def **process\_scroll**(event):

print(*"mouse:"*, event.step)

fig, ax = plt.subplots(1, 1)

fig.canvas.mpl\_connect(*'scroll\_event'*, process\_scroll)

plt.show()

上滚：

mouse: 1.0

下滚：

mouse: -1.0

### 坐标事件

#### 坐标进入、离开事件

Code:

import matplotlib.pyplot as plt

global ax1,ax2,ax3,ax4

def **process\_axes\_enter**(event):

ax = event.inaxes

if ax is ax1:

print(*'Enter ax1'*)

elif ax is ax2:

print(*'Enter ax2'*)

elif ax is ax3:

print(*'Enter ax3'*)

elif ax is ax4:

print(*'Enter ax4'*)

def **process\_axes\_leave**(event):

ax = event.inaxes

if ax is ax1:

print(*'leave ax1'*)

elif ax is ax2:

print(*'leave ax2'*)

elif ax is ax3:

print(*'leave ax3'*)

elif ax is ax4:

print(*'leave ax4'*)

fig = plt.figure()

ax1 = fig.add\_subplot(2,2,1)

ax2 = fig.add\_subplot(2,2,2)

ax3 = fig.add\_subplot(2,2,3)

ax4 = fig.add\_subplot(2,2,4)

fig.canvas.mpl\_connect(*'axes\_enter\_event'*, process\_axes\_enter)

fig.canvas.mpl\_connect(*'axes\_leave\_event'*, process\_axes\_leave)

plt.show()

输出：

Enter ax2

leave ax2

Enter ax1

leave ax1

Enter ax4

leave ax4

Enter ax3

leave ax3

#### 坐标的缩放与平移

### pick事件

#### Demo1

Code:

import numpy as np

import matplotlib.pyplot as plt

fig,ax = plt.subplots(1,1)

ax.plot(np.arange(10), *'o'*, **picker**=5) # 5 points tolerance

def **on\_pick**(event):

line = event.artist

xdata, ydata = line.get\_data()

ind = event.ind

print(*'on pick line:'*, np.array([xdata[ind], ydata[ind]]).T)

cid = fig.canvas.mpl\_connect(*'pick\_event'*, on\_pick)

plt.show()

一次点击1~6个点的输出：

on pick line: [[ 1. 1.]]

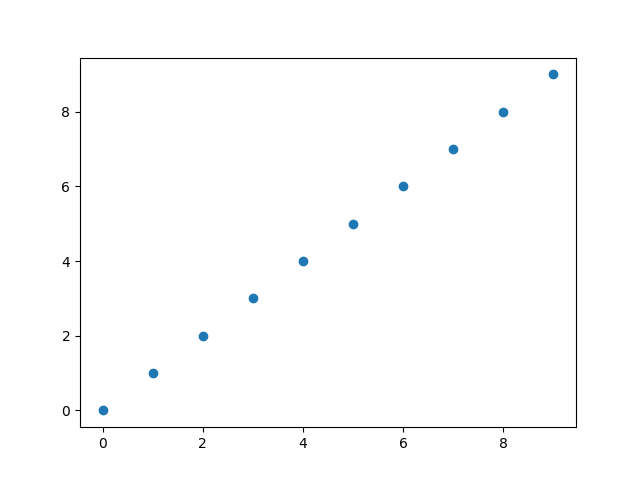
on pick line: [[ 2. 2.]]

on pick line: [[ 3. 3.]]

on pick line: [[ 4. 4.]]

on pick line: [[ 5. 5.]]

on pick line: [[ 6. 6.]]



代码分析：

ax.plot(np.arange(10), *'o'*, picker=5) # 5 points tolerance

其中参数picker表示最大可接受的误差。

Debug进入函数on\_pick,通过type和dir命令查看event的类型及event包含的属性：

>>> type(event)

<class 'matplotlib.backend\_bases.PickEvent'>

>>> dir(event)

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', 'artist', 'canvas', 'guiEvent', 'ind', 'mouseevent', 'name']

查看当前按下的鼠标键值（此处为左键）

>>> event.mouseevent.button

1

查看当前pick的点对应的index（此处为3）：

>>> event.ind

array([3], dtype=int32)

查看当前artist的类型:

>>> type(event.artist)

<class 'matplotlib.lines.Line2D'>

继续查看artist的属性(非常多)：

>>> dir(event.artist)

['\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getstate\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', '\_agg\_filter', '\_alpha', '\_animated', '\_antialiased', '\_axes', '\_clipon', '\_clippath', '\_color', '\_contains', '\_dashOffset', '\_dashSeq', '\_dashcapstyle', '\_dashjoinstyle', '\_drawStyles\_l', '\_drawStyles\_s', '\_draw\_dash\_dot', '\_draw\_dashed', '\_draw\_dotted', '\_draw\_lines', '\_draw\_solid', '\_draw\_steps\_mid', '\_draw\_steps\_post', '\_draw\_steps\_pre', '\_drawstyle', '\_get\_markerfacecolor', '\_get\_rgba\_face', '\_get\_rgba\_ln\_color', '\_get\_transformed\_path', '\_gid', '\_invalidx', '\_invalidy', '\_is\_sorted', '\_label', '\_lineStyles', '\_linestyle', '\_linestyles', '\_linewidth', '\_marker', '\_markeredgecolor', '\_markeredgewidth', '\_markerfacecolor', '\_markerfacecoloralt', '\_markersize', '\_markevery', '\_mouseover', '\_oid', '\_path', '\_path\_effects', '\_picker', '\_prop\_order', '\_propobservers', '\_rasterized', '\_remove\_method', '\_set\_gc\_clip', '\_sketch', '\_snap', '\_solidcapstyle', '\_solidjoinstyle', '\_split\_drawstyle\_linestyle', '\_stale', '\_sticky\_edges', '\_subslice', '\_transform', '\_transformSet', '\_transform\_path', '\_transformed\_path', '\_url', '\_us\_dashOffset', '\_us\_dashSeq', '\_visible', '\_x', '\_x\_filled', '\_xcid', '\_xorig', '\_xy', '\_y', '\_ycid', '\_yorig', 'add\_callback', 'aname', 'axes', 'clipbox', 'contains', 'convert\_xunits', 'convert\_yunits', 'draw', 'drawStyleKeys', 'drawStyles', 'eventson', 'figure', 'fillStyles', 'filled\_markers', 'findobj', 'format\_cursor\_data', 'get\_aa', 'get\_agg\_filter', 'get\_alpha', 'get\_animated', 'get\_antialiased', 'get\_axes', 'get\_c', 'get\_children', 'get\_clip\_box', 'get\_clip\_on', 'get\_clip\_path', 'get\_color', 'get\_contains', 'get\_cursor\_data', 'get\_dash\_capstyle', 'get\_dash\_joinstyle', 'get\_data', 'get\_drawstyle', 'get\_figure', 'get\_fillstyle', 'get\_gid', 'get\_label', 'get\_linestyle', 'get\_linewidth', 'get\_ls', 'get\_lw', 'get\_marker', 'get\_markeredgecolor', 'get\_markeredgewidth', 'get\_markerfacecolor', 'get\_markerfacecoloralt', 'get\_markersize', 'get\_markevery', 'get\_mec', 'get\_mew', 'get\_mfc', 'get\_mfcalt', 'get\_ms', 'get\_path', 'get\_path\_effects', 'get\_picker', 'get\_pickradius', 'get\_rasterized', 'get\_sketch\_params', 'get\_snap', 'get\_solid\_capstyle', 'get\_solid\_joinstyle', 'get\_transform', 'get\_transformed\_clip\_path\_and\_affine', 'get\_url', 'get\_visible', 'get\_window\_extent', 'get\_xdata', 'get\_xydata', 'get\_ydata', 'get\_zorder', 'have\_units', 'hitlist', 'ind\_offset', 'is\_dashed', 'is\_figure\_set', 'is\_transform\_set', 'lineStyles', 'markers', 'mouseover', 'pchanged', 'pick', 'pickable', 'pickradius', 'properties', 'recache', 'recache\_always', 'remove', 'remove\_callback', 'set', 'set\_aa', 'set\_agg\_filter', 'set\_alpha', 'set\_animated', 'set\_antialiased', 'set\_axes', 'set\_c', 'set\_clip\_box', 'set\_clip\_on', 'set\_clip\_path', 'set\_color', 'set\_contains', 'set\_dash\_capstyle', 'set\_dash\_joinstyle', 'set\_dashes', 'set\_data', 'set\_drawstyle', 'set\_figure', 'set\_fillstyle', 'set\_gid', 'set\_label', 'set\_linestyle', 'set\_linewidth', 'set\_ls', 'set\_lw', 'set\_marker', 'set\_markeredgecolor', 'set\_markeredgewidth', 'set\_markerfacecolor', 'set\_markerfacecoloralt', 'set\_markersize', 'set\_markevery', 'set\_mec', 'set\_mew', 'set\_mfc', 'set\_mfcalt', 'set\_ms', 'set\_path\_effects', 'set\_picker', 'set\_pickradius', 'set\_rasterized', 'set\_sketch\_params', 'set\_snap', 'set\_solid\_capstyle', 'set\_solid\_joinstyle', 'set\_transform', 'set\_url', 'set\_visible', 'set\_xdata', 'set\_ydata', 'set\_zorder', 'stale', 'stale\_callback', 'sticky\_edges', 'update', 'update\_from', 'validCap', 'validJoin', 'verticalOffset', 'zorder']

#### 事件

**mpl\_connect**(event\_name, call\_back\_func)

其中event\_name包括:

*- 'button\_press\_event'*

*- 'button\_release\_event'*

*- 'draw\_event'*

*- 'key\_press\_event'*

*- 'key\_release\_event'*

*- '**motion\_notify\_event'*

*- 'pick\_event'*

*- 'resize\_event'*

*- 'scroll\_event'*

*- 'figure\_enter\_event',*

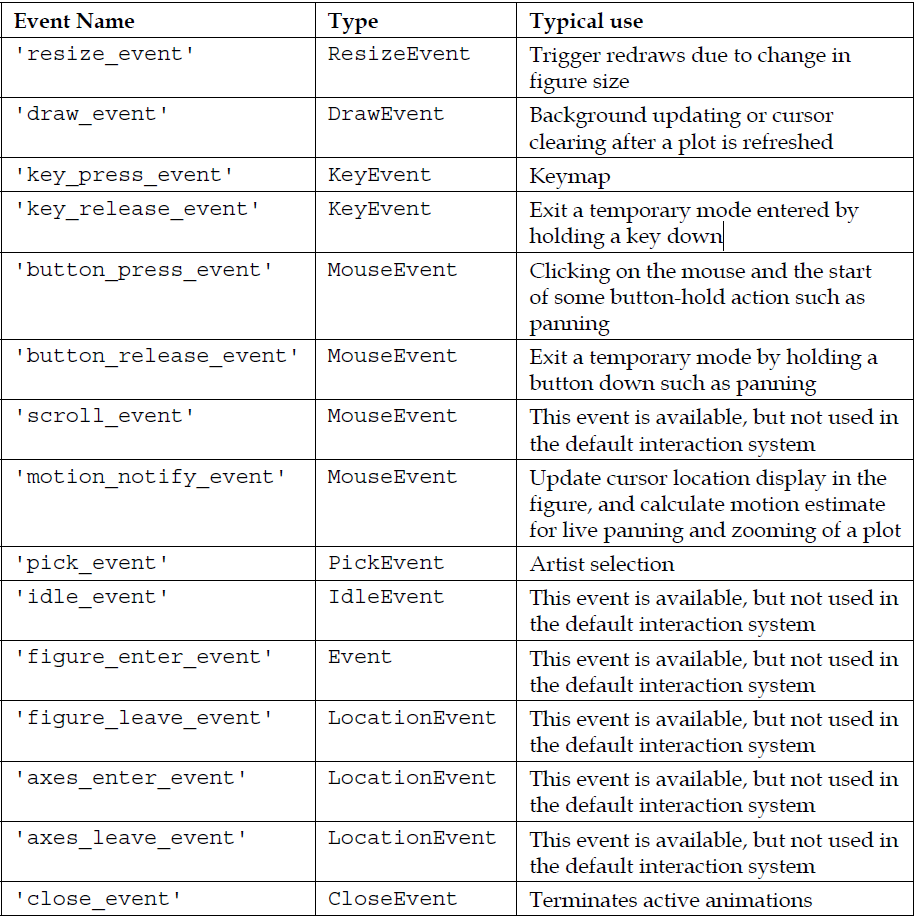
*- 'figure\_leave\_event',*

*- 'axes\_enter\_event',*

*- 'axes\_leave\_event'*

*- 'close\_event'*

其中，各个事件代表的意义如下：



#### Demo2

Code:

#encoding=utf-8

import numpy as np

import matplotlib.pyplot as plt

line\_list = []

def **pick\_event**(event):

obj = event.artist

for l in line\_list:

line = l[0]

if obj is line:

line.set\_alpha(1.0)

else:

line.set\_alpha(0.3)

def **legend\_demo1**():

*'''*

*pick envet: 实现方式一*

*'''*

fig = plt.figure()

ax = fig.add\_subplot(111)

x = np.linspace(0, np.pi\*2, 360)

y1 = np.sin(x)

y2 = np.cos(x)

line1 = ax.plot(x,y1,*'r'*, label=*'sin(x)'*, alpha=0.3)

line2 = ax.plot(x,y2,*'g'*, label=*'cos(x)'*, alpha=0.3)

line1[0].set\_picker(True)

line2[0].set\_picker(5)

line\_list.append(line1)

line\_list.append(line2)

legd = ax.legend(loc=*'best'*)

fig.canvas.mpl\_connect(*'pick\_event'*, pick\_event)

plt.show()

def **legend\_demo2**():

*'''*

*pick envet: 实现方式二*

*'''*

fig = plt.figure()

ax = fig.add\_subplot(111)

x = np.linspace(0, np.pi\*2, 360)

y1 = np.sin(x)

y2 = np.cos(x)

line1 = ax.plot(x,y1,*'r'*, label=*'sin(x)'*, alpha=0.3, picker=True)

line2 = ax.plot(x,y2,*'g'*, label=*'cos(x)'*, alpha=0.3, picker=5)

line\_list.append(line1)

line\_list.append(line2)

legd = ax.legend(loc=*'best'*)

fig.canvas.mpl\_connect(*'pick\_event'*, pick\_event)

plt.show()

def **main**():

#legend\_demo1()

legend\_demo2()

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

## 控件

### 内置控件

Matplotlib的内置控件都在widgets模块中，我们通过dir命令查看一下widgets模块都包含哪些属性(类)：

>>> from matplotlib import widgets

>>> dir(widgets)

['AxesWidget', 'Button', 'CheckButtons', 'Circle', 'Cursor', 'Ellipse', 'Ellipse

Selector', 'Lasso', 'LassoSelector', 'Line2D', 'LockDraw', 'MultiCursor', 'Radio

Buttons', 'Rectangle', 'RectangleSelector', 'Slider', 'SpanSelector', 'SubplotTo

ol', 'ToolHandles', 'Widget', '\_SelectorWidget', '\_\_builtins\_\_', '\_\_cached\_\_', '

\_\_doc\_\_', '\_\_file\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'abso

lute\_import', 'blended\_transform\_factory', 'copy', 'dist', 'division', 'np', 'pr

int\_function', 'six', 'unicode\_literals', 'zip']

#### Button

Code:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.widgets import Button

freqs = np.arange(2, 20, 3)

fig, ax = plt.subplots()

plt.subplots\_adjust(bottom=0.2)

t = np.arange(0.0, 1.0, 0.001)

s = np.sin(2\*np.pi\*freqs[0]\*t)

l, = plt.plot(t, s, lw=2)

class **Index**(object):

ind = 0

def **next**(*self*, event):

*self*.ind += 1

i = *self*.ind % len(freqs)

ydata = np.sin(2\*np.pi\*freqs[i]\*t)

l.set\_ydata(ydata)

plt.draw()

def **prev**(*self*, event):

*self*.ind -= 1

i = *self*.ind % len(freqs)

ydata = np.sin(2\*np.pi\*freqs[i]\*t)

l.set\_ydata(ydata)

plt.draw()

callback = Index()

axprev = plt.axes([0.7, 0.05, 0.1, 0.075])

axnext = plt.axes([0.81, 0.05, 0.1, 0.075])

bnext = Button(axnext, *'Next'*)

bnext.on\_clicked(callback.next)

bprev = Button(axprev, *'Previous'*)

bprev.on\_clicked(callback.prev)

plt.show()

#### CheckButtons

Code：

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.widgets import CheckButtons

t = np.arange(0.0, 2.0, 0.01)

s0 = np.sin(2\*np.pi\*t)

s1 = np.sin(4\*np.pi\*t)

s2 = np.sin(6\*np.pi\*t)

fig, ax = plt.subplots()

l0, = ax.plot(t, s0, visible=False, lw=2)

l1, = ax.plot(t, s1, lw=2)

l2, = ax.plot(t, s2, lw=2)

plt.subplots\_adjust(left=0.2)

rax = plt.axes([0.05, 0.4, 0.1, 0.15])

check = CheckButtons(rax, (*'2 Hz'*, *'4 Hz'*, *'6 Hz'*), (False, True, True))

def **func**(label):

if label == *'2 Hz'*:

l0.set\_visible(not l0.get\_visible())

elif label == *'4 Hz'*:

l1.set\_visible(not l1.get\_visible())

elif label == *'6 Hz'*:

l2.set\_visible(not l2.get\_visible())

plt.draw()

check.on\_clicked(func)

plt.show()

#### RadioButtons

Code:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.widgets import RadioButtons

t = np.arange(0.0, 2.0, 0.01)

s0 = np.sin(2\*np.pi\*t)

s1 = np.sin(4\*np.pi\*t)

s2 = np.sin(8\*np.pi\*t)

fig, ax = plt.subplots()

l, = ax.plot(t, s0, lw=2, color=*'red'*)

plt.subplots\_adjust(left=0.3)

axcolor = *'lightgoldenrodyellow'*

rax = plt.axes([0.05, 0.7, 0.15, 0.15], facecolor=axcolor)

radio = RadioButtons(rax, (*'2 Hz'*, *'4 Hz'*, *'8 Hz'*))

def **hzfunc**(label):

hzdict = {*'2 Hz'*: s0, *'4 Hz'*: s1, *'8 Hz'*: s2}

ydata = hzdict[label]

l.set\_ydata(ydata)

plt.draw()

radio.on\_clicked(hzfunc)

rax = plt.axes([0.05, 0.4, 0.15, 0.15], facecolor=axcolor)

radio2 = RadioButtons(rax, (*'red'*, *'blue'*, *'green'*))

def **colorfunc**(label):

l.set\_color(label)

plt.draw()

radio2.on\_clicked(colorfunc)

rax = plt.axes([0.05, 0.1, 0.15, 0.15], facecolor=axcolor)

radio3 = RadioButtons(rax, (*'-'*, *'--'*, *'-.'*, *'steps'*, *':'*))

def **stylefunc**(label):

l.set\_linestyle(label)

plt.draw()

radio3.on\_clicked(stylefunc)

plt.show()

#### Slider

Code：

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.widgets import Slider, Button, RadioButtons

fig, ax = plt.subplots()

plt.subplots\_adjust(left=0.25, bottom=0.25)

t = np.arange(0.0, 1.0, 0.001)

a0 = 5

f0 = 3

s = a0\*np.sin(2\*np.pi\*f0\*t)

l, = plt.plot(t, s, lw=2, color=*'red'*)

plt.axis([0, 1, -10, 10])

axcolor = *'lightgoldenrodyellow'*

axfreq = plt.axes([0.25, 0.1, 0.65, 0.03], facecolor=axcolor)

axamp = plt.axes([0.25, 0.15, 0.65, 0.03], facecolor=axcolor)

sfreq = Slider(axfreq, *'Freq'*, 0.1, 30.0, valinit=f0)

samp = Slider(axamp, *'Amp'*, 0.1, 10.0, valinit=a0)

def **update**(val):

amp = samp.val

freq = sfreq.val

l.set\_ydata(amp\*np.sin(2\*np.pi\*freq\*t))

fig.canvas.draw\_idle()

sfreq.on\_changed(update)

samp.on\_changed(update)

resetax = plt.axes([0.8, 0.025, 0.1, 0.04])

button = Button(resetax, *'Reset'*, color=axcolor, hovercolor=*'0.975'*)

def **reset**(event):

sfreq.reset()

samp.reset()

button.on\_clicked(reset)

rax = plt.axes([0.025, 0.5, 0.15, 0.15], facecolor=axcolor)

radio = RadioButtons(rax, (*'red'*, *'blue'*, *'green'*), active=0)

def **colorfunc**(label):

l.set\_color(label)

fig.canvas.draw\_idle()

radio.on\_clicked(colorfunc)

plt.show()

#### SpanSelector

Code:

*"""*

*The SpanSelector is a mouse widget to select a xmin/xmax range and plot the*

*detail view of the selected region in the lower axes*

*"""*

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.widgets import SpanSelector

fig = plt.figure(figsize=(8, 6))

ax = fig.add\_subplot(211, facecolor=*'#FFFFCC'*)

x = np.arange(0.0, 5.0, 0.01)

y = np.sin(2\*np.pi\*x) + 0.5\*np.random.randn(len(x))

ax.plot(x, y, *'-'*)

ax.set\_ylim(-2, 2)

ax.set\_title(*'Press left mouse button and drag to test'*)

ax2 = fig.add\_subplot(212, facecolor=*'#FFFFCC'*)

line2, = ax2.plot(x, y, *'-'*)

def **onselect**(xmin, xmax):

indmin, indmax = np.searchsorted(x, (xmin, xmax))

indmax = min(len(x) - 1, indmax)

thisx = x[indmin:indmax]

thisy = y[indmin:indmax]

line2.set\_data(thisx, thisy)

ax2.set\_xlim(thisx[0], thisx[-1])

ax2.set\_ylim(thisy.min(), thisy.max())

fig.canvas.draw()

# set useblit True on gtkagg for enhanced performance

span = SpanSelector(ax, onselect, *'horizontal'*, useblit=True,

rectprops=dict(alpha=0.5, facecolor=*'red'*))

plt.show()

### Embedding Qt

#### 示例

Code:

import sys

import matplotlib

from PyQt5.QtWidgets import QWidget, QApplication, QVBoxLayout, QMainWindow

from PyQt5 import QtCore

matplotlib.use(*'Qt5Agg'*)

import matplotlib.pyplot as plt

import numpy as np

from matplotlib.backends.backend\_qt5agg import FigureCanvas

class **MainWindow**(QMainWindow):

def **\_\_init\_\_**(*self*, parent=None):

super(MainWindow, *self*).\_\_init\_\_()

*self*.sutupMainWindow()

*self*.draw()

def **sutupMainWindow**(*self*):

*self*.main\_widget = QWidget(*self*)

*self*.\_\_layout\_\_ = QVBoxLayout(*self*.main\_widget)

*self*.setLayout(*self*.\_\_layout\_\_)

*self*.mainfigure = plt.figure()

*self*.subfigure = plt.figure()

*self*.mainCanvas = *self*.mainfigure.canvas

*self*.subCanvas = *self*.subfigure.canvas

*self*.\_\_layout\_\_.addWidget(*self*.mainCanvas)

*self*.\_\_layout\_\_.addWidget(*self*.subCanvas)

*self*.main\_widget.setLayout(*self*.\_\_layout\_\_)

*self*.setCentralWidget(*self*.main\_widget)

def **draw**(*self*):

*self*.ax1 = *self*.mainfigure.add\_subplot(111)

*self*.ax2 = *self*.subfigure.add\_subplot(111)

theta = np.linspace(start=0, stop=2\*np.pi, num=360, endpoint=True)

r = 2

x = r\*np.cos(theta)

y = r\*np.sin(theta)

*self*.ax1.plot(x,y, *'r'*)

*self*.ax2.plot(x,y, *'g'*)

*self*.ax1.set\_aspect(1)

*self*.ax2.set\_aspect(1)

def **main**():

app = QApplication(sys.argv)

win = MainWindow()

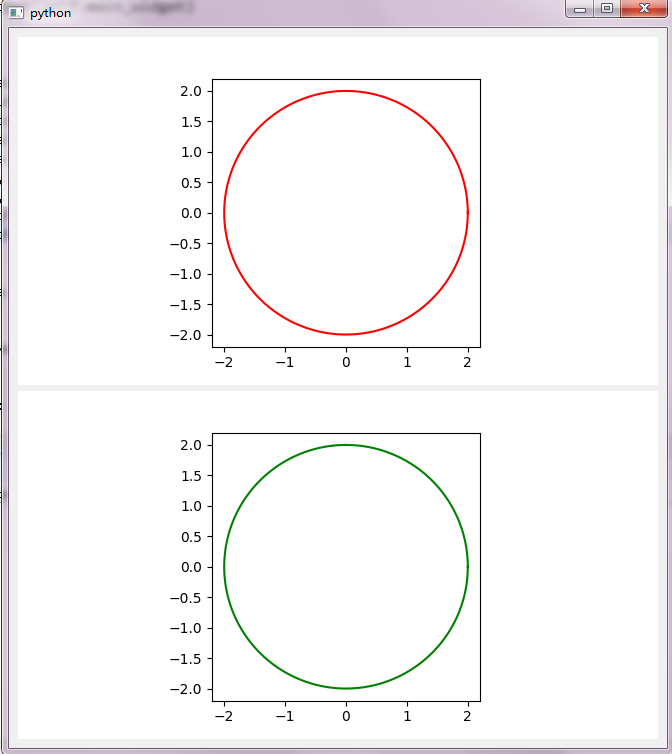
win.show()

return sys.exit(app.exec\_())

if \_\_name\_\_ == *'\_\_main\_\_'*:

main()

输出：



#### 使用Qt Designer

# Finance

# 3D