

□□

```
import matplotlib.pyplot as plt
```

□□□

□□ plt.scatter(x,y) □□□□□

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

x = [0.13,0.22,0.39,0.59,0.68,0.74,0.93]
y = [0.75,0.34,0.44,0.52,0.80,0.25,0.65]

plt.figure(figsize = (8,6))
## □□□□
plt.scatter(x,y,marker = 's',s = 50)

## □□□□□□
for x,y in zip(x,y):
    plt.annotate('(%s,%s)'%(x,y),xy=(x,y),xytext = {0,-15},textcoords = 'offset
points')

plt.show()
```

□□□

□□ plt.hist() □□□□

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

data = np.random.normal(0,20,1000)
bins = np.arange(-100,100,5)

plt.hist(data,bins = bins)
plt.show()
```

□□□

- □□ plt.bar(x,y) □□□□
- □□ plt.barh(x,y) □□□□□□□

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

np.random.seed(0)
x = np.arange(5)
```

```
y = np.random.randn(5)
fig,axes = plt.subplots(ncols = 2)
## 亂数生成
v_bars = axes[0].bar(x,y,color = 'red',alpha = 0.5)
h_bars = axes[1].barh(x,y,color = 'red',alpha = 0.5)
```

3D

3D

```
from mpl_toolkits.mplot3d import Axes3d
```

3D

```
ax = plt.gca(projection = '3d')
```

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

t = np.linspace(0,10,100)
x = np.sin(t)
y = np.cos(t)
z = t

## 3Dプロット用ax
ax = plt.gca(projection = '3d')
## 3D用ax
ax.plot(x,y,z,label = '3D')
ax.set_xlabel('x label')
ax.set_ylabel('y label')
ax.set_zlabel('z label')
ax.legend()

plt.show()
```

API

- `plt.plot(x,y)` :
◦ label
◦ color
◦ linestyle
◦ linewidth
◦ alpha
◦ marker
- `plt.subplot(nrows,ncols,index)` :
◦ nrows
◦ ncols
◦ index
◦ ==plt.plot()==

- `plt.subplots()` : 生成一个包含子图的 figure，返回 axes 对象

-

- `nrows` [n̄r̄w̄s̄]
 - `ncols` [n̄k̄l̄s̄]
 - `sharex(y)` [sh̄r̄ēk̄ x̄(ȳ)]

- axes

- `plot()` 亂子圖
 - `set_title()` 亂題目
 - `set_x(y)label()` 亂軸標
 - `set_x(y)lim()` 亂範圍

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

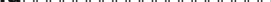
np.random.seed(0)
x = np.arange(5)
y = np.random.randn(5)
fig,axes = plt.subplots(ncols = 2)

axes[0].plot(x,y,label = '1')
axes[1].plot(y,x,label = '2')

plt.show()
```

- `plt.show()` :

- `pyplot.figure(figsize=(20,8),dpi=80)` :  
dpi  
figsize

-  **fig** 

- `fig.add_subplot`

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

```
画图
fig = plt.figure(figsize=(8, 4), dpi=100)

坐标轴
ax1 = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(1, 2, 2)

数据
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)

绘制
ax1.plot(x, y1, label='Sine')
ax1.plot(x, y2, label='Cosine')
ax2.plot(x, y1, label='Sine')
ax2.plot(x, y2, label='Cosine')

显示
plt.show()
```

```

ax1.set_title('Sine Wave')
ax1.legend()

ax2.plot(x, y2, label='Cosine', color='orange')
ax2.set_title('Cosine Wave')
ax2.legend()

显示
plt.show()

```

- `pyplot.savefig("filename")` :`svg`文件名
- `pyplot.x(y)ticks()` :`x(y)轴`,`刻度范围`
  - `xticks(ticks,labels,**kwargs)`
    - `ticks`:`x轴刻度值`
    - `labels`:`x轴刻度值``ticks`与`labels`一一对应
    - `**kwargs`参数
      - `rotation`:`旋转`

```

import matplotlib.pyplot as plt
x轴
bar_label = ['bar1','bar2','bar3']
x_pos = list(range(len(bar_label)))
plt.xticks(x_pos,bar_label)

```

- `font_manager``fontproperties`对象`my_font`
- `pyplot.x(y)label()` :`x(y)轴`
  - `font_size`
- `pyplot.title()` :`标题`
  - `字体`
- `pyplot.grid(alpha)` :`网格`
  - `alpha`
- `pyplot.legend()` :`图例`
  - `loc`:`位置`
- `plt.annotate(text,xy,xytext=None,arrowprops=None,**kwargs)`
  - `text`
  - `xy`:`标注点``(x,y)`
  - `xytext`:`箭头指向点``(x,y)`
  - `arrowprops`:`箭头属性`

- \*\*kwargs : ディクショナリオブジェクト

```
• import matplotlib.pyplot as plt

plt.annotate('Local Max', xy=(1.57, 1), xytext=(3, 0.5),
 arrowprops=dict(facecolor='red', shrink=0.05),
 fontsize=12, color='blue')
```

- plt.fill\_between(x, y1, y2=0, \*\*kwargs) : 複数の範囲を塗りつぶす

- x : x 軸の値
- y1 : y 軸の下限
- y2 : y 軸の上限 (y2=y1 の場合は y=0)
- \*\*kwargs : 色 (color)、透明度 (alpha)、ラベル (label) など

- plt.xlim() : x軸の範囲

```
• import matplotlib.pyplot as plt
1から10までのx軸
plt.xlim(1,10)
```

- plt.vlines(x,ymin,ymax,colors,linestyles) : 直線

- x : 直線のx座標
- ymin : 直線の下限
- ymax : 直線の上限
- colors : 直線の色
- linestyles : 直線のスタイル

```
• import matplotlib.pyplot as plt

x = [1,2,3,4,5]
y = [2,3,1,4,7]
plt.plot(x,y)

x=3でy=0からy=5までの直線
plt.vlines(x = 3,ymin = 0,ymax = 5,colors = 'r',linestyles='dash')
plt.show()
```

- zip(x,y) : 二つのリストを組み合わせる