

matplotlib

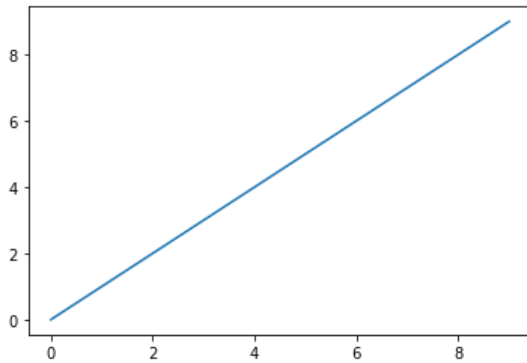
```
In [1]: import matplotlib.pyplot as plt  
import numpy as np
```

```
In [2]: %matplotlib inline
```

plt.plot(data, color, linestyle, marker, label, drawstyle)

```
In [3]: data = np.arange(10)  
plt.plot(data)
```

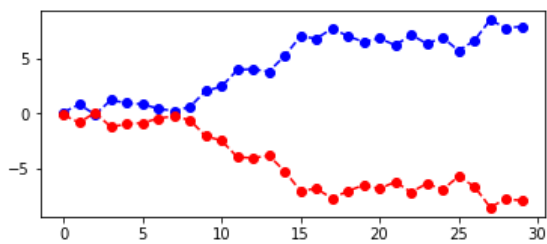
```
Out[3]: [matplotlib.lines.Line2D at 0x11dba8e1e50]
```



color, linestyle, marker, label : 颜色, 标记, 线型, 图例

plt.subplots(nrows, ncols, sharex, sharey): 调整图片大小

```
In [4]: data = np.random.randn(30).cumsum()  
plt.plot(data, color='b', linestyle='--', marker='o')  
plt.plot(-data, 'ro--')  
plt.subplots_adjust(top=0.6)
```



drawstyle :

plt.legend(loc): 设置图例

```
In [5]: data = np.random.randn(30).cumsum()
plt.plot(data, 'k--', label='Default')
plt.plot(data, 'k-', drawstyle='steps-post', label='steps-post')
plt.legend(loc='best')
```

Out[5]: <matplotlib.legend.Legend at 0x11dbb0d9850>

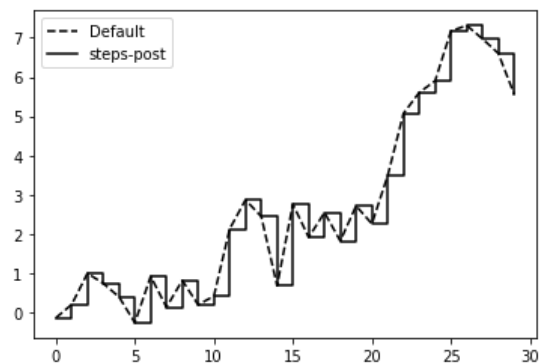


Figure 和 Subplot

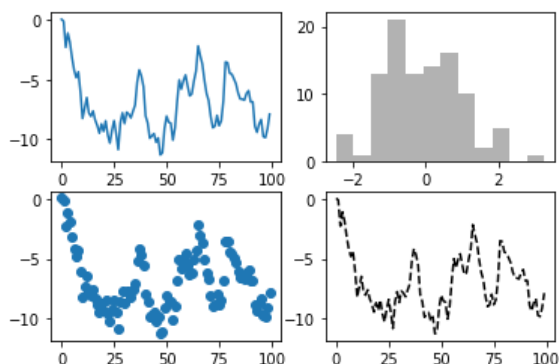
`plt.figure(figsize)` : 创建一个的 Figure 对象; matplotlib 的图像都位于 Figure 对象中

```
In [6]: fig = plt.figure() #创建一个的 Figure 对象
```

<Figure size 432x288 with 0 Axes>

`fig.add_subplot(nrows, ncols, index)` : 创建一个的 AxesSubplot 对象, 位于 `[nrows, ncols]` 的 `index` 位置

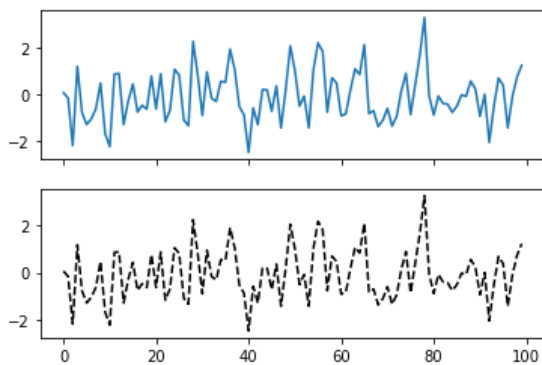
```
In [7]: fig = plt.figure() #创建一个的 Figure 对象
ax1 = fig.add_subplot(2, 2, 1)
ax2 = fig.add_subplot(2, 2, 2)
ax3 = fig.add_subplot(2, 2, 3)
ax4 = fig.add_subplot(2, 2, 4)
data = np.random.randn(100)
plt.plot(data.cumsum(), 'k--')
ax1.plot(data.cumsum())
ax2.hist(data, bins=12, color='k', alpha=0.3) # 统计直方图
ax3.scatter(np.arange(100), data.cumsum()) # 散点图
plt.savefig('normal.png', dpi=1080, bbox_inches='tight')
```



`plt.subplots(nrows, ncols, sharex, sharey)` : 创建 Figure 对象和 一组 AxesSubplot 对象 (存放于 ndarray 中)

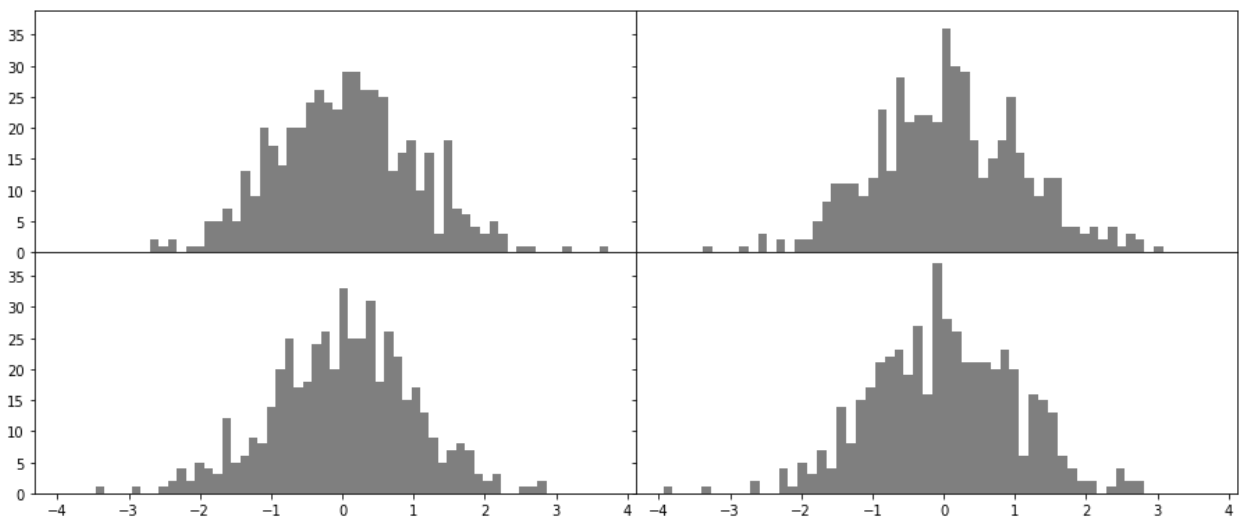
```
In [8]: fig, axes = plt.subplots(2, 1, sharex='all', sharey='all') # 共享x轴, y轴
# fig : Figure 对象 ; axes : ndarray 对象, 用于索引 AxesSubplot
plt.plot(data, 'k--')
axes[0].plot(data) #对图片进行索引
```

```
Out[8]: [matplotlib.lines.Line2D at 0x11dbb291610>]
```



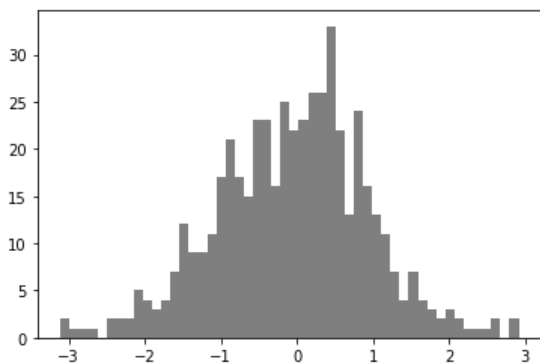
`plt.subplots_adjust(left, bottom, right, top, wspace, hspace)` : 调整 subplot 大小和间距

```
In [9]: fig, axes = plt.subplots(2, 2, sharex=True, sharey=True)
for i in range(2):
    for j in range(2):
        axes[i, j].hist(np.random.randn(500), bins=50, color='k', alpha=0.5)
plt.subplots_adjust(left=0, bottom=0, right=2, top=1.2,
                    wspace=0, hspace=0)
```



刻度, 标签 和 图例

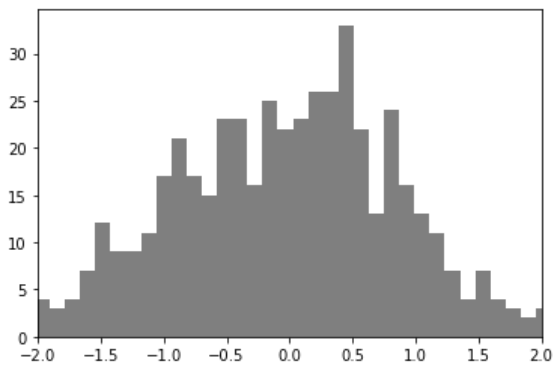
```
In [10]: data = np.random.randn(500)
_ = plt.hist(data, bins=50, color='k', alpha=0.5)
```



`plt.xlim()`, `plt.xlim([start, stop])` : 控制绘图范围

```
In [11]: plt.hist(data, bins=50, color='k', alpha=0.5)
plt.xlim(), plt.xlim([-2, 2])
```

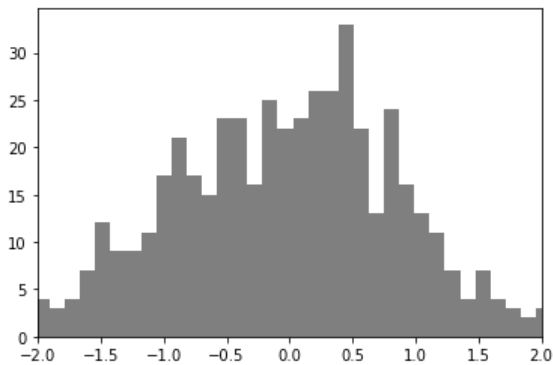
```
Out[11]: ((-3.411006317547548, 3.221768778645931), (-2.0, 2.0))
```



`ax.get_xlim()`, `ax.set_xlim([start, stop])` : 控制绘图范围

```
In [12]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5)
ax.get_xlim(), ax.set_xlim([-2, 2])
```

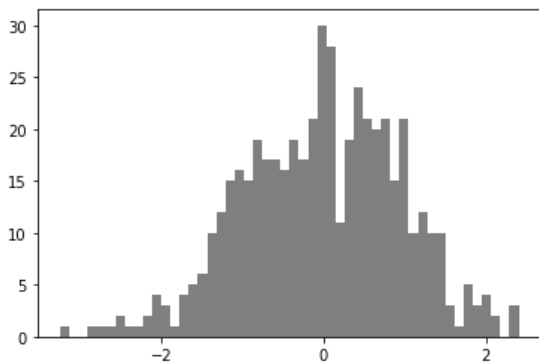
```
Out[12]: ((-3.411006317547548, 3.221768778645931), (-2.0, 2.0))
```



`plt.xticks()`, `plt.xticks(ticks, labels)` : 控制刻度位置和标签

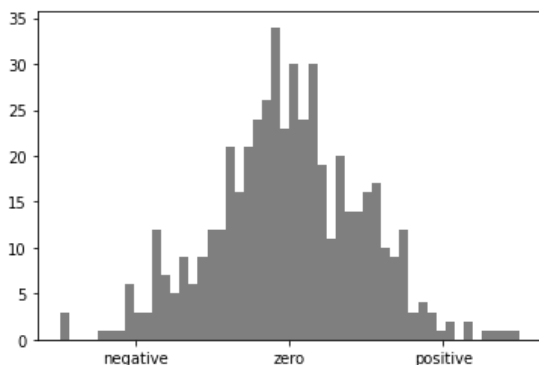
```
In [13]: data = np.random.randn(500)
plt.hist(data, bins=50, color='k', alpha=0.5)
plt.xticks(ticks = [-2, 0, 2])
```

```
Out[13]: ([<matplotlib.axis.XTick at 0x11dbb949430>,
<matplotlib.axis.XTick at 0x11dbb949400>,
<matplotlib.axis.XTick at 0x11dbb937f40>],
[Text(0, 0, ''), Text(0, 0, ''), Text(0, 0, '')])
```



```
In [14]: data = np.random.randn(500)
plt.hist(data, bins=50, color='k', alpha=0.5)
plt.xticks(ticks = [-2, 0, 2], labels=['negative', 'zero', 'positive'])
```

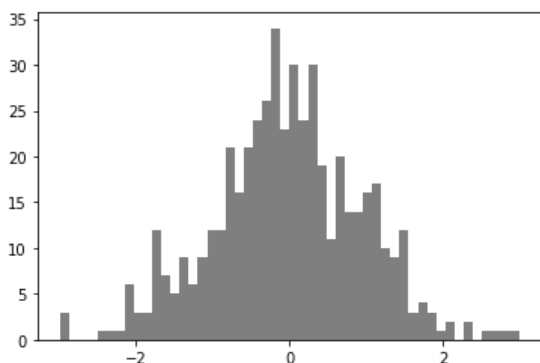
```
Out[14]: ([<matplotlib.axis.XTick at 0x11dbba06310>,
<matplotlib.axis.XTick at 0x11dbba062e0>,
<matplotlib.axis.XTick at 0x11dbb8735e0>],
[Text(-2, 0, 'negative'), Text(0, 0, 'zero'), Text(2, 0, 'positive')])
```



`ax.get_xticks()`, `ax.set_xticks(ticks)` : 控制刻度位置

```
In [15]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5)
ax.get_xticks(), ax.set_xticks([-2, 0, 2])
```

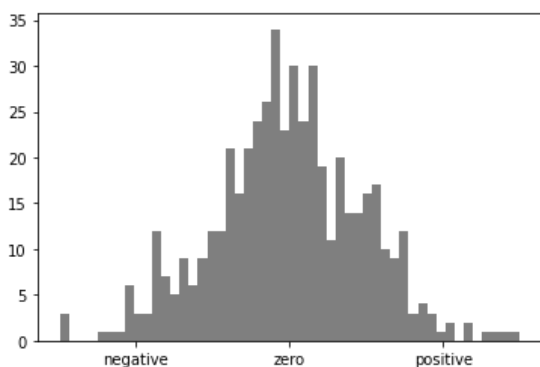
```
Out[15]: (array([-4., -3., -2., -1.,  0.,  1.,  2.,  3.,  4.]),
[<matplotlib.axis.XTick at 0x11dbbac5760>,
 <matplotlib.axis.XTick at 0x11dbbac5730>,
 <matplotlib.axis.XTick at 0x11dbbabe040>])
```



`ax.get_xticklabels()`, `ax.set_xticklabels(list)` : 控制刻度标签

```
In [16]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5)
ax.set_xticks([-2, 0, 2])
ax.set_xticklabels(['negative', 'zero', 'positive'])
```

```
Out[16]: [Text(-2, 0, 'negative'), Text(0, 0, 'zero'), Text(2, 0, 'positive')]
```

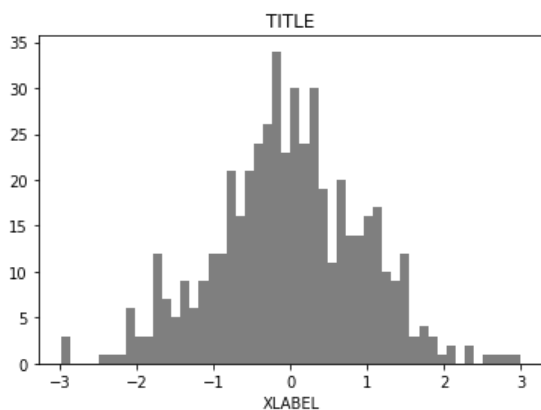


标题 和 轴标签

`ax.set_xlabel(xlabel)`, `ax.set_title(title)`

```
In [17]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5)
ax.set_xlabel('XLABEL'), ax.set_title('TITLE')
```

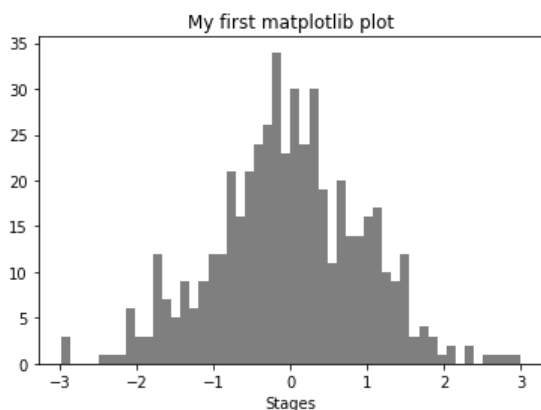
```
Out[17]: (Text(0.5, 0, 'XLABEL'), Text(0.5, 1.0, 'TITLE'))
```



`ax.set(\kwargs)`**

```
In [18]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5)
props = {
    'title': 'My first matplotlib plot',
    'xlabel': 'Stages'
}
ax.set(**props)
```

```
Out[18]: [Text(0.5, 1.0, 'My first matplotlib plot'), Text(0.5, 0, 'Stages')]
```

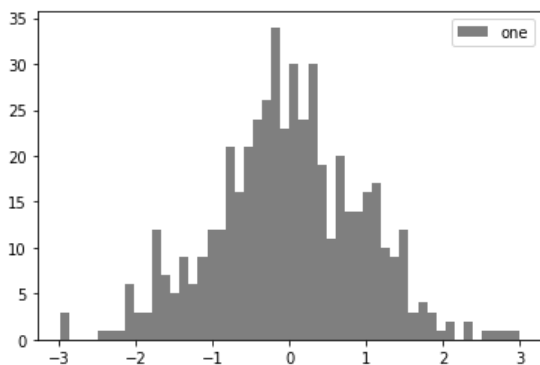


图例

`ax.legend(loc)`

```
In [19]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5, label='one')
ax.legend(loc='best')
```

Out[19]: <matplotlib.legend.Legend at 0x11dbcdba2e0>



添加文本

`ax.text(x, y, str, rotation, ha, va, fontdict = dict(...), bbox = dict(...))`

`fontdict = dict(fontsize, color, family, weight)`

`bbox = dict(facecolor, edgecolor, edgewidth, alpha, pad)`

`rotation`: 旋转, **`ha`**: 左右对齐方式, **`va`**: 上下对齐方式

`fontdict`: 字体属性字典

`family`: 字体, **`weight`**: 磅值

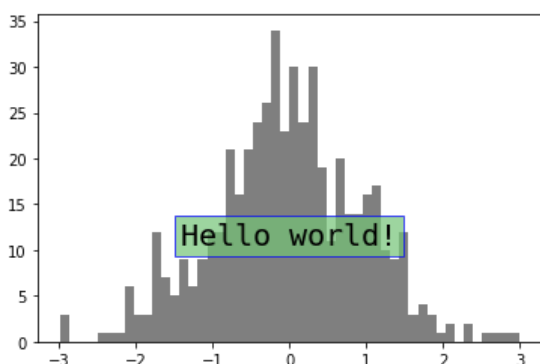
`bbox`: 边框填充属性字典

`facecolor`: 填充色, **`edgecolor`**: 边框色, **`edgewidth`**: 边框粗细

`alpha`: 透明度, **`pad`**: 文本与框周围距离, **`boxstyle`**: 形状参数

```
In [20]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5, label='one')
ax.text(0, 10, 'Hello world!', ha = 'center', va = 'bottom',
        fontdict = dict(family='monospace', fontsize=20),
        bbox = dict(facecolor='#74c476', edgecolor='b', alpha=0.7))
```

Out[20]: Text(0, 10, 'Hello world!')



`ax.annotate(text, xy, xytext, xycoords, textcoords, ha, va, weight, color, bbox, arrowprops)`

`xy = (x, y)`

`xytext = (x, y)`

`bbox = dict(facecolor, edgecolor, edgewidth, alpha, pad)`

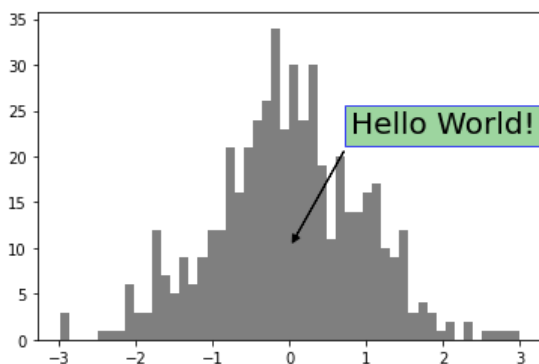
`arrowprops = dict(facecolor, hatch, alpha, shrink, width, headwidth, headlength)`

xy: 箭头指向位置, ***xytext***: 文本位置
xycoords: 箭头位置的参考坐标系, ***textcoords***: 选择文本位置的参考坐标系
ha: 文本左右对齐方式, ***va***: 文本上下对齐方式
weight: 文本磅值, ***va***: 文本颜色
bbox: 文本边框填充属性字典

arrowprops: 箭头属性字典
facecolor: 填充色, ***hatch***: 填充形状
shrink: 两端收缩控制, ***width***: 箭身宽度
headwidth: 箭头宽度, ***headlength***: 箭头长度

```
In [21]: fig, ax = plt.subplots(1, 1)
ax.hist(data, bins=50, color='k', alpha=0.5, label='one')
ax.annotate('Hello World!', (0, 10), (2, 25), ha='center', va='top', fontsize=20,
           arrowprops=dict(facecolor='k', width=0.1, headwidth=5, headlength=5, shrink=0.05),
           bbox=dict(facecolor='#74c476', edgecolor='b', alpha=0.7)
           )
```

Out[21]: Text(2, 25, 'Hello World!')



添加图形

ax.add_patch(shp)

plt.Rectangle(xy, width, height, angle, color, alpha)

plt.Circle(xy, r, color, alpha)

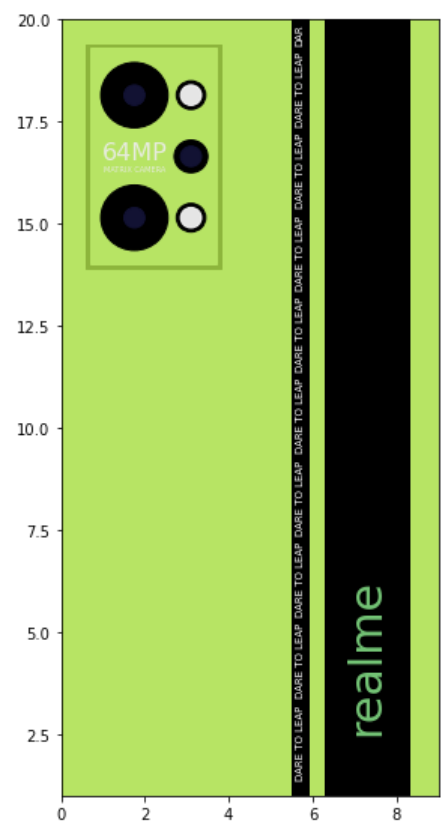
plt.Polygon([xy1, xy2, ...], color, alpha)

```

In [22]: fig = plt.figure(figsize=(4.5, 9.5))
ax = fig.add_subplot(1, 1, 1)
ax.set_xlim(0,9)
ax.set_ylim(1,20)
# 正方形
rect = plt.Rectangle((0, 1), 9, 20, color='#b7e464', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((0.6, 13.9), 3.2, 5.5, color='#8db53c', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((0.7, 14), 3, 5.3, color='#b7e464', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((6.3, 0), 2, 20, color='k', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((5.5, 0), 0.4, 20, color='k', alpha=1)
ax.add_patch(rect)
# 圓形
circ = plt.Circle((1.75, 15.15), 0.8, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((1.75, 15.15), 0.25, color='#121233', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((1.75, 18.15), 0.8, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((1.75, 18.15), 0.25, color='#121233', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 16.65), 0.4, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 16.65), 0.25, color='#121233', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 15.15), 0.35, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 18.15), 0.35, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 15.15), 0.25, color='w', alpha=0.9)
ax.add_patch(circ)
circ = plt.Circle((3.1, 18.15), 0.25, color='w', alpha=0.9)
ax.add_patch(circ)
ax.text(7.3, 2.5, 'realme', ha = 'center', va = 'bottom', rotation=90,
        fontdict = dict(fontsize=30, color='#74c476'))
for i in range(0, 9):
    ax.text(5.7, 1.4+i*2, 'DARE TO LEAP', ha = 'center', va = 'bottom', rotation=90,
            fontdict = dict(fontsize=7, color='w'))
ax.text(5.7, 1.4+(i+1)*2, 'DAR', ha = 'center', va = 'bottom', rotation=90,
        fontdict = dict(fontsize=7, color='w'))
ax.text(1.75, 16.75, '64MP', ha = 'center', va = 'center', alpha=0.9,
        fontdict = dict(fontsize=16, color='EEEEEE', weight='light'))
ax.text(1.75, 16.35, 'MATRIX CAMERA', ha = 'center', va = 'center', alpha=0.9,
        fontdict = dict(fontsize=5, color='EEEEEE', weight='light'))

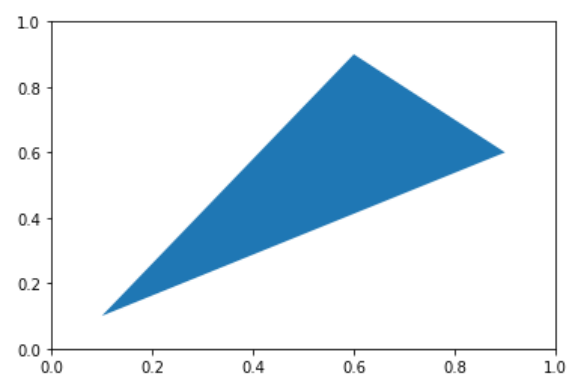
```

Out[22]: Text(1.75, 16.35, 'MATRIX CAMERA')



```
In [23]: fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
pgon = plt.Polygon([[0.1, 0.1], [0.6, 0.9], [0.9, 0.6]])
ax.add_patch(pgon)
```

Out[23]: <matplotlib.patches.Polygon at 0x11dbb80d640>



保存图片

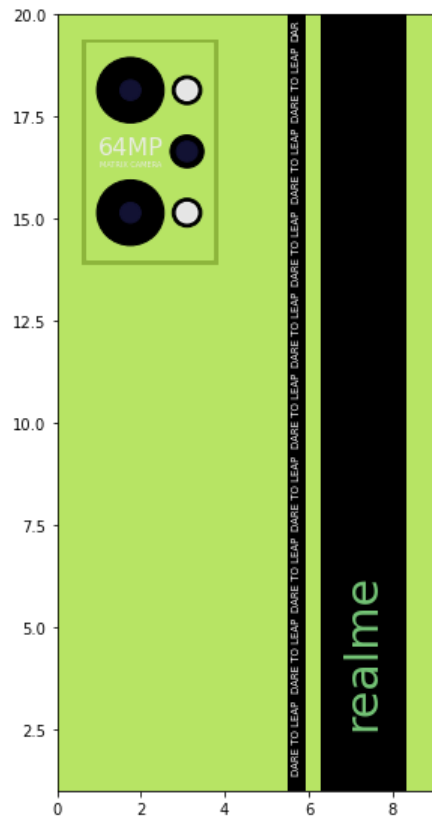
`plt.savefig(fname, dpi, bbox_inches)`

`dpi` : 每英寸点数分辨率, default: 100
`bbox_inches` : 控制图片周围白边, `bbox_inches = 'tight'`

```

In [24]: fig = plt.figure(figsize=(4.5, 9.5))
ax = fig.add_subplot(1, 1, 1)
ax.set_xlim(0,9)
ax.set_ylim(1,20)
# 正方形
rect = plt.Rectangle((0, 1), 9, 20, color='#b7e464', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((0.6, 13.9), 3.2, 5.5, color='#8db53c', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((0.7, 14), 3, 5.3, color='#b7e464', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((6.3, 0), 2, 20, color='k', alpha=1)
ax.add_patch(rect)
rect = plt.Rectangle((5.5, 0), 0.4, 20, color='k', alpha=1)
ax.add_patch(rect)
# 圓形
circ = plt.Circle((1.75, 15.15), 0.8, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((1.75, 15.15), 0.25, color='#121233', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((1.75, 18.15), 0.8, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((1.75, 18.15), 0.25, color='#121233', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 16.65), 0.4, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 16.65), 0.25, color='#121233', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 15.15), 0.35, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 18.15), 0.35, color='k', alpha=1)
ax.add_patch(circ)
circ = plt.Circle((3.1, 15.15), 0.25, color='w', alpha=0.9)
ax.add_patch(circ)
circ = plt.Circle((3.1, 18.15), 0.25, color='w', alpha=0.9)
ax.add_patch(circ)
ax.text(7.3, 2.5, 'realme', ha = 'center', va = 'bottom', rotation=90,
        fontdict = dict(fontsize=30, color='#74c476'))
for i in range(0, 9):
    ax.text(5.7, 1.4+i*2, 'DARE TO LEAP', ha = 'center', va = 'bottom', rotation=90,
            fontdict = dict(fontsize=7, color='w'))
ax.text(5.7, 1.4+(i+1)*2, 'DAR', ha = 'center', va = 'bottom', rotation=90,
        fontdict = dict(fontsize=7, color='w'))
ax.text(1.75, 16.75, '64MP', ha = 'center', va = 'center', alpha=0.9,
        fontdict = dict(fontsize=16, color='EEEEEE', weight='light'))
ax.text(1.75, 16.35, 'MATRIX CAMERA', ha = 'center', va = 'center', alpha=0.9,
        fontdict = dict(fontsize=5, color='EEEEEE', weight='light'))
plt.savefig('realmeneo2.png', dpi=1080, bbox_inches='tight')

```



matplotlib 的全局配置

`plt.rc(group, \kwargs)**` : 全局参数的定义

`group` : `figure`, `font`, `axes`, `xtick`, `grid`, `legend` , 希望定义的对象

```
In [25]: # plt.rc('figure', figsize=(4, 3))
# font_options = {
#     'family' : 'monospace',
#     'weight' : 'bold',
#     'size' : 10}
# plt.rc('font', **font_options)
```

```
In [26]: fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.text(0.2, 0.2, 'Hello world!', ha = 'left', va = 'bottom',
        fontdict=dict(fontsize=20),
        bbox = dict(facecolor='#74c476', edgecolor='b', alpha=0.7))
```

Out[26]: Text(0.2, 0.2, 'Hello world!')

