

Tham Khảo Nhanh Matplotlib

Figures, axes, plots, tùy chỉnh

Cơ Bản

Thiết Lập

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

fig, ax = plt.subplots()
ax.plot([1, 2, 3], [4, 5, 6])
plt.show()
```

Hai Phong Cách

```
plt.plot(x, y)  Pyplot (nhanh, không tưởng minh)
ax.plot(x, y)  OO API (tưởng minh, khuyến nghị)
```

Figure & Axes

Tạo Figure

```
fig = plt.figure(figsize=(8, 6), dpi=100)
fig, ax = plt.subplots()
fig, axes = plt.subplots(2, 3, figsize=(12, 8))
fig, (ax1, ax2) = plt.subplots(1, 2)
```

Subplot Layout

```
fig, axes = plt.subplots(2, 2, sharex=True, sharey=True)
plt.tight_layout() # adjust spacing
plt.subplots_adjust(hspace=0.4, wspace=0.3)
```

Thuộc Tính Axes

```
ax.set_title('Title')      Đặt tiêu đề
ax.set_xlabel('x') / set_ylabel('y')  Nhãn trục
ax.set_xlim(0, 10)        Giới hạn trục x
ax.set_ylim(-1, 1)        Giới hạn trục y
ax.legend()                Hiển thị chú thích
ax.grid(True)              Hiển thị lưới
ax.set_aspect('equal')     Tỷ lệ trục bằng nhau
```

Các Loại Đồ Thị

Đồ Thị Đường

```
ax.plot(x, y)
ax.plot(x, y, 'r--', linewidth=2, label='sin')
ax.plot(x, y, color='#2ca02c', marker='o', markersize=5)
```

Scatter Plot

```
ax.scatter(x, y)
ax.scatter(x, y, c=colors, s=sizes, alpha=0.7,
           cmap='viridis')
```

Biểu Đồ Thanh

```
ax.bar(categories, values)
ax.bar(x, y, width=0.4, color='steelblue')
ax.barh(categories, values) # horizontal
```

Biểu Đồ Tần Số

```
ax.hist(data, bins=30)
ax.hist(data, bins=30, density=True, alpha=0.7)
```

Các Loại Đồ Thị Khác

```
ax.pie(sizes, labels=labels)  Biểu đồ tròn
ax.boxplot(data)              Box plot
ax.violinplot(data)           Violin plot
ax.imshow(img, cmap='gray')   Hiển thị ảnh hoặc ma trận 2D
ax.contour(X, Y, Z)           Đường đồng mức
ax.contourf(X, Y, Z)          Đường đồng mức tô màu
ax.errorbar(x, y, yerr=e)     Đồ thị có thanh sai số
ax.fill_between(x, y1, y2)    Tô màu giữa hai đường
ax.stackplot(x, y1, y2)       Đồ thị xếp chồng
ax.step(x, y)                 Đồ thị bậc thang
```

3D Plots

Đồ Thị 3D

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis')
ax.scatter(x, y, z)
```

Tùy Chỉnh

Màu Sắc & Kiểu Đường

```
'r' / 'b' / 'g'           Màu: đỏ / xanh dương / xanh lá
'-' / '--' / ':' / '-.'   Kiểu đường: liền / đứt / chấm / gạch-chấm
'o' / 's' / '^' / 'x'     Marker: tròn / vuông / tam giác / chéo
color='#hex' / 'name'     Màu tùy chỉnh theo hex hoặc tên
alpha=0.7                 Độ trong suốt (0-1)
```

Văn Bản & Chú Thích

```
ax.text(x, y, 'label', fontsize=12, ha='center')
ax.annotate('peak', xy=(x, y),
            xytext=(x+1, y+0.5),
            arrowprops=dict(arrowstyle='->'))
```

Tick & Spine

```
ax.set_xticks([0, 1, 2, 3])
ax.set_xticklabels(['a', 'b', 'c', 'd'])
ax.tick_params(axis='x', rotation=45)
ax.spines['top'].set_visible(False)
```

Colormap & Colorbar

```
sc = ax.scatter(x, y, c=vals, cmap='plasma')
fig.colorbar(sc, ax=ax, label='Value')

# Common colormaps:
# viridis, plasma, inferno, magma (sequential)
# RdBu, seismic (diverging)
# tab10, Set1 (categorical)
```

Style & Theme

Sử Dụng Styles

```
plt.style.use('seaborn-v0_8')
plt.style.use('ggplot')
plt.style.use('dark_background')
print(plt.style.available) # list styles
```

Cấu Hình Toàn Cục

```
plt.rcParams['font.size'] = 12
plt.rcParams['figure.figsize'] = (8, 6)
plt.rcParams['axes.spines.top'] = False
```

Lưu & Xuất

Lưu Figure

```
fig.savefig('plot.png', dpi=150, bbox_inches='tight')
fig.savefig('plot.pdf', format='pdf')
fig.savefig('plot.svg')
```

Backend

```
plt.show()          Hiển thị cửa sổ tương tác
plt.close()         Đóng figure hiện tại
plt.close('all')    Đóng tất cả figures
%matplotlib inline Jupyter: hiển thị inline
%matplotlib widget Jupyter: hiển thị tương tác
```

Mẫu Phổ Biến

Nhiều Đường Trong Một Trục

```
x = np.linspace(0, 2*np.pi, 100)
ax.plot(x, np.sin(x), label='sin')
ax.plot(x, np.cos(x), label='cos')
ax.legend()
```

Subplots Với Dữ Liệu

```
fig, axes = plt.subplots(1, 2, figsize=(10, 4))
axes[0].plot(x, np.sin(x)); axes[0].set_title('Sin')
axes[1].plot(x, np.cos(x)); axes[1].set_title('Cos')
plt.tight_layout()
```

Histogram + KDE

```
import scipy.stats as stats
ax.hist(data, bins=30, density=True, alpha=0.6)
kde = stats.gaussian_kde(data)
ax.plot(x, kde(x), 'r-', lw=2)
```