

Matplotlib 快速参考

图形、坐标轴、绘图类型与自定义

基础绘图

折线图

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 8, 3]
plt.plot(x, y)
plt.show()
```

快捷绘图

```
plt.plot(y) # x auto 0..N-1
plt.plot(x, y, "ro--") # red circles, dashed
plt.plot(x, y, "bs-") # blue squares, solid
```

格式字符串代码

'r'	'g'	'b'	'k'	红、绿、蓝、黑
'o'	's'	'^'	'D'	圆形、方形、三角形、菱形标记
'-'	'--'	'-.'	':'	实线、虚线、点划线、点线

子图

Figure 与 Axes

```
fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_title("Single Plot")
plt.show()
```

网格子图

```
fig, axes = plt.subplots(2, 2, figsize=(8, 6))
axes[0, 0].plot(x, y)
axes[0, 1].bar(x, y)
axes[1, 0].scatter(x, y)
fig.tight_layout()
```

共享坐标轴

```
fig, (ax1, ax2) = plt.subplots(1, 2,
                              sharey=True, figsize=(10, 4))
ax1.plot(x, y)
ax2.plot(x, y2)
```

标签与标题

坐标轴标签与标题

```
plt.plot(x, y)
plt.xlabel("Time (s)")
plt.ylabel("Value")
plt.title("Sensor Reading")
plt.show()
```

面向对象风格

```
fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_xlabel("X"); ax.set_ylabel("Y")
ax.set_title("My Plot")
```

注释

```
ax.annotate("Peak", xy=(4, 8),
            xytext=(3, 9),
            arrowprops=dict(arrowstyle="->"))
```

自定义

颜色与样式

```
plt.plot(x, y, color="#FF5733",
         linewidth=2, linestyle="--")
plt.plot(x, y2, color="steelblue",
         marker="o", markersize=5)
```

图形大小与 DPI

```
fig, ax = plt.subplots(figsize=(10, 6), dpi=100)
plt.rcParams["figure.figsize"] = (8, 5)
```

样式表

```
print(plt.style.available) # list all
plt.style.use("seaborn-v0_8")
plt.style.use("ggplot")
```

柱状图与直方图

柱状图

```
labels = ["A", "B", "C", "D"]
values = [23, 45, 12, 37]
plt.bar(labels, values, color="teal")
plt.show()
```

分组 / 堆叠柱状图

```
import numpy as np
x = np.arange(4); w = 0.35
plt.bar(x - w/2, v1, w, label="2024")
plt.bar(x + w/2, v2, w, label="2025")
plt.xticks(x, labels)
```

直方图

```
data = np.random.randn(1000)
plt.hist(data, bins=30, edgecolor="black",
         alpha=0.7)
plt.show()
```

散点图与折线图

散点图

```
plt.scatter(x, y, c="red", s=50,
           alpha=0.6, edgecolors="black")
plt.show()
```

带色图的散点图

```
sc = plt.scatter(x, y, c=values,
                 cmap="viridis", s=sizes)
plt.colorbar(sc, label="Intensity")
```

多条折线

```
plt.plot(x, y1, label="Train")
plt.plot(x, y2, label="Validation")
plt.legend()
plt.show()
```

坐标轴与刻度

坐标轴范围与比例

```
ax.set_xlim(0, 10)
ax.set_ylim(-1, 1)
ax.set_xscale("log")
ax.set_yscale("log")
```

自定义刻度

```
ax.set_xticks([0, 1, 2, 3, 4])
ax.set_xticklabels(["Mon", "Tue", "Wed",
                  "Thu", "Fri"], rotation=45)
```

网格线

```
ax.grid(True, linestyle="--", alpha=0.5)
ax.grid(axis="y") # horizontal only
```

图例

添加图例

```
ax.plot(x, y, label="Series A")
ax.plot(x, y2, label="Series B")
ax.legend(loc="upper right")
```

图例位置

'best'	自动最佳位置 (默认)
'upper left'	左上角
'lower right'	右下角
'center'	坐标轴中心
'bbox_to_anchor=(1, 1)'	放置在坐标轴外部

图例自定义

```
ax.legend(fontsize=8, frameon=False,
         ncol=2, title="Legend")
```

保存

保存到文件

```
plt.savefig("plot.png", dpi=300,
         bbox_inches="tight")
plt.savefig("plot.pdf")
plt.savefig("plot.svg", transparent=True)
```

支持的格式

PNG	光栅图, 适合网页/屏幕
PDF	矢量图, 适合打印/论文
SVG	矢量图, 可缩放, 适合网页
EPS	矢量图, 旧式科学期刊

从 Figure 对象保存

```
fig, ax = plt.subplots()
ax.plot(x, y)
fig.savefig("output.png", dpi=150,
         facecolor="white")
```

常见模式

双 Y 轴

```
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.plot(x, temp, "r-", label="Temp")
ax2.plot(x, pressure, "b-", label="Pressure")
```

区域填充

```
ax.fill_between(x, y_low, y_high,
               alpha=0.3, color="blue")
```

Matplotlib 快速参考

用 imshow 绘制热力图

```
data = np.random.rand(10, 10)
plt.imshow(data, cmap="hot",
            interpolation="nearest")
plt.colorbar()
```

饼图

```
plt.pie(sizes, labels=labels,
        autopct="%1.1f%%", startangle=90)
plt.axis("equal")
```