Referencia Rápida de Matplotlib

Figuras, ejes, gráficas y personalización

Gráficas básicas

Gráfica de línea

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

Atajos de gráfica rápida

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

Códigos de formato de cadena

`r` `g` `b` `k`	Rojo, verde, azul, negro
`o` `s` `^` `D`	Marcadores: círculo, cuadrado, triángulo, diamante
`-` `--` `-.` `:`	Líneas: continua, discontinua, mixta, punteada

Subgráficas

Figura y ejes

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

Cuadrícula de subgráficas

```
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()
```

Ejes compartidos

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

Etiquetas y títulos

Etiquetas de ejes y título

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

Etiquetas estilo OO

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

Anotaciones

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

Personalización

Colores y estilos

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

Tamaño de figura y DPI

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

Hojas de estilo

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

Barras e histogramas

Gráfico de barras

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

Barras agrupadas / apiladas

```
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)
```

Histograma

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

Dispersión y líneas

Diagrama de dispersión

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

Dispersión con mapa de color

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

Múltiples líneas

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

Ejes y marcas

Límites y escala de ejes

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

Marcas personalizadas

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

Cuadrícula

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

Leyendas

Agregar leyendas

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

Posición de la leyenda

`'best'`	Mejor posición automática (por defecto)
`'upper left'`	Esquina superior izquierda
`'lower right'`	Esquina inferior derecha
`'center'`	Centro de los ejes
`bbox_to_anchor=(1, 1)`	Colocar fuera del área de ejes

Personalizar la leyenda

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

Guardar

Guardar en archivo

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

Formatos soportados

PNG	Raster, mejor para web/pantalla
PDF	Vectorial, mejor para impresión/artículos
SVG	Vectorial, escalable para web
EPS	Vectorial, revistas científicas heredadas

Guardar desde objeto figura

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

Patrones comunes

Ejes dobles (dos escalas Y)

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

Relleno entre líneas

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

Mapa de calor con imshow

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

Gráfico circular

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