

Matplotlib Riferimento Rapido

Figure, assi, grafici e personalizzazione

Grafici di Base

Grafico a Linee

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

Scorciatoie per Grafici Rapidi

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

Codici della Stringa di Formato

<code>'r'</code>	<code>'g'</code>	<code>'b'</code>	<code>'k'</code>	Rosso, verde, blu, nero
<code>'o'</code>	<code>'s'</code>	<code>'^'</code>	<code>'D'</code>	Marker: cerchio, quadrato, triangolo, diamante
<code>'-'</code>	<code>'--'</code>	<code>'-.'</code>	<code>'.'</code>	Linee: continua, tratteggiata, tratto-punto, punteggiata

Sottografici

Figure e Assi

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

Griglia di Sottografici

```
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()
```

Assi Condivisi

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

Etichette e Titoli

Etichette degli Assi e Titolo

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

Etichette in Stile OO

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

Annotazioni

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

Personalizzazione

Colori e Stili

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

Dimensione della Figura e DPI

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

Fogli di Stile

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

Barre e Istogramma

Grafico a Barre

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

Barre Raggruppate / Impilate

```
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)
```

Istogramma

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

Dispersione e Linee

Grafico a Dispersione

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

Dispersione con Mappa Colori

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

Linee Multiple

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

Assi e Tacche

Limiti e Scala degli Assi

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

Tacche Personalizzate

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

Griglia

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

Legende

Aggiunta di Legende

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

Posizionamento della Legenda

<code>'best'</code>	Posizione migliore automatica (predefinita)
<code>'upper left'</code>	Angolo in alto a sinistra
<code>'lower right'</code>	Angolo in basso a destra
<code>'center'</code>	Centro degli assi
<code>'bbox_to_anchor=(1, 1)'</code>	Posiziona fuori dall'area degli assi

Personalizzazione della Legenda

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

Salvataggio

Salva su File

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

Formati Supportati

PNG	Raster, ideale per web/schermo
PDF	Vettoriale, ideale per stampa/articoli
SVG	Vettoriale, scalabile per il web
EPS	Vettoriale, riviste scientifiche legacy

Salva dall'Oggetto Figure

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

Pattern Comuni

Assi Doppi (Due Scale Y)

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

Riempimento tra Curve

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

Heatmap con imshow

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

Matplotlib Riferimento Rapido

Grafico a Torta

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