

Référence rapide Matplotlib

Figures, axes, tracés et personnalisation

Tracés de base

Tracé en lignes

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

Raccourcis de tracé rapides

```
plt.plot(y) # x auto 0..N-1
plt.plot(x, y, "ro-") # ronds rouges, tirets
plt.plot(x, y, "bs-") # carrés bleus, plein
```

Codes de chaîne de format

```
`r` `g` `b` `k` Rouge, vert, bleu, noir
`o` `s` `^` `D` Marqueurs : rond, carré, triangle, losange
`-` `--` `.` `:` Lignes : plein, tirets, tiret-point, pointillés
```

Sous-graphiques

Figure et axes

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

Grille de sous-graphiques

```
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()
```

Axes partagés

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

Étiquettes et titres

Étiquettes d'axes et titre

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

Style orienté objet

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

Annotations

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

Personnalisation

Couleurs et styles

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

Taille de figure et DPI

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

Feuilles de style

```
print(plt.style.available) # lister toutes
plt.style.use("seaborn-v0_8")
plt.style.use("ggplot")
```

Barres et histogrammes

Diagramme en barres

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

Barres groupées / empilées

```
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)
```

Histogramme

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

Nuages de points et lignes

Nuage de points

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

Nuage de points avec palette de couleurs

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

Lignes multiples

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

Axes et graduations

Limites et échelle des axes

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

Graduations personnalisées

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

Grille

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

Légendes

Ajouter des légendes

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

Placement de la légende

```
`best` Meilleure position automatique (défaut)
`upper left` Coin supérieur gauche
`lower right` Coin inférieur droit
`center` Centre des axes
`bbox_to_anchor=(1, 1)` Placer en dehors de la zone des axes
```

Personnalisation de la légende

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

Enregistrement

Enregistrer dans un fichier

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

Formats pris en charge

PNG Matriciel, idéal pour le web/écran
PDF Vectoriel, idéal pour l'impression/articles
SVG Vectoriel, scalable pour le web
EPS Vectoriel, revues scientifiques classiques

Enregistrer depuis l'objet figure

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

Modèles courants

Axes doubles (deux échelles Y)

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

Remplissage entre courbes

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

Carte de chaleur avec imshow

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

Diagramme circulaire

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