

Matplotlib

Romain HÉRAULT

Introduction

Matplotlib est la librairie de graphiques scientifiques du projet scipy.

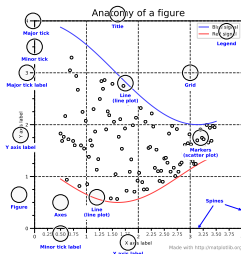


FIGURE – Exemple disponible sur

<http://matplotlib.org/examples/showcase/anatomy.html>

Utilisation

```
import matplotlib
matplotlib.use('pdf')
```

Deux API différentes : Pyplot 1/2

pyplot

API identique à celle d'octave/matlab.

```
import matplotlib.pyplot as plt
plt.figure()
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.savefig('pyplotintro.pdf')
```

Elle utilise le contexte (non réentrante).

Deux API différentes : Artist 2/2

Artist

API propre à Matplotlib

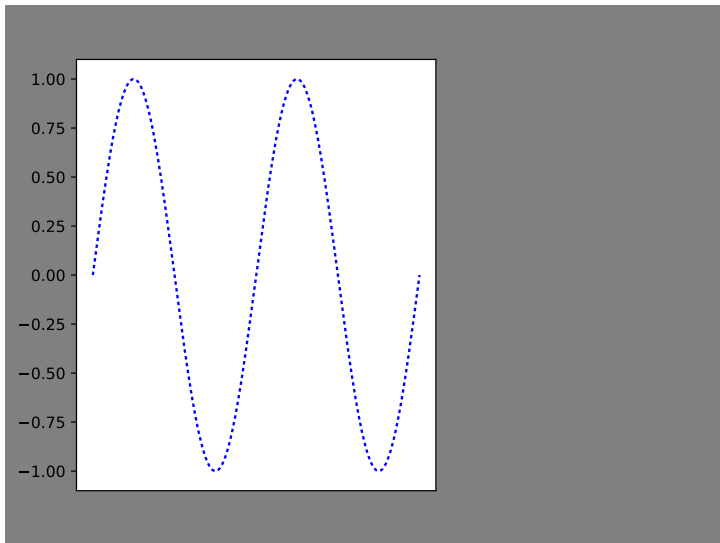
```
from matplotlib.figure import Figure
from matplotlib.backends.backend_pdf import FigureCanvasPdf as
    FigureCanvas
fig = Figure()
ax = fig.add_subplot(111)
ax.plot([1, 2, 3, 4])
ax.set_ylabel('some numbers')
canvas = FigureCanvas(fig)
canvas.draw()
fig.savefig('arstistintro.pdf')
```

Tout est objet. Plus complexe mais plus puissante.

Classes usuelles d'Artist

- `matplotlib.backend_bases.FigureCanvasBase`
Classe abstraite dépend du backend chargée des primitives de bas niveau de dessin et d'entrée/sortie.
- `matplotlib.figure.Figure`
Conteneur des autres objets Artist.
- `matplotlib.axes.Axes`
Emplacement pour un graphe, plusieurs **Axes** possibles par **Figure**.
- `matplotlib.axis.Axis`
Un axe de coordonnées d'un **Axes**
- `matplotlib.lines.Line2D`
Ensemble de vecteurs 2D
- `matplotlib.text.Text`
Gère les éléments textuels présent sur les graphes.

Exemple : dessin



Exemple : code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

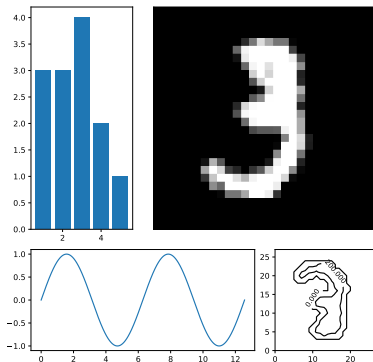
import matplotlib
matplotlib.use('pdf')
import matplotlib.pyplot as plt
import numpy as np

nt = 1000
t = np.linspace(0, 4 * np.pi, nt)
x = np.sin(t)

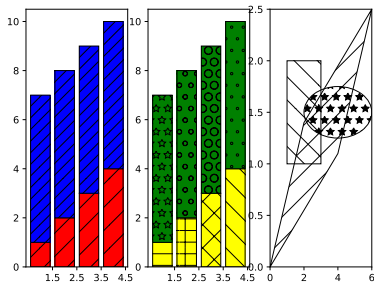
aFig = plt.figure() # gere automatiquement canvas et figure
anAxes = aFig.add_axes([0.1, 0.1, 0.5, 0.8])
aXaxis = anAxes.get_xaxis()
aXaxis.set_visible(False)
aLine, = anAxes.plot(t, x, color='blue')
aLine.set_linestyle(":")
aFig.savefig('exemple.pdf', facecolor='grey')
```

Exercice 1 : Gridspec

- `matplotlib.gridspec.Gridspec`
- `matplotlib.figure.Figure.add_subplot`
- `matplotlib.axes.Axes.bar`
- `matplotlib.axes.Axes.imshow`
- `matplotlib.axes.Axes.contour`
- `matplotlib.axes.Axes.clabel`

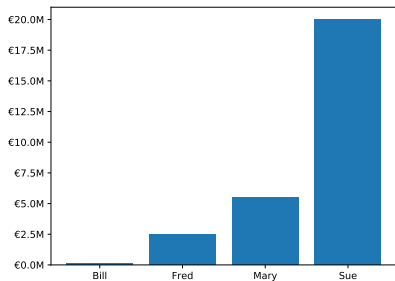


Exercice 2 : hatch



- `matplotlib.axes.Axes.set_xticks`
- `matplotlib.axes.Axes.set_xlim`
- `matplotlib.axes.Axes.set_ylim`
- `matplotlib.axes.Axes.add_patch`
- `matplotlib.axes.Axes.fill`
- `matplotlib.patches.Ellipse`
- `matplotlib.patches.Polygon`
- `matplotlib.patches.Rectangle.set_hatch`

Exercice 3 : formatter



- `matplotlib.ticker.FuncFormatter`
- `matplotlib.axes.Axes.get_yaxis`
- `matplotlib.axes.Axes.set_xticklabels`
- `matplotlib.axis.Axis.set_major_formatter`