# Graphe simple de sinus et cosinus

On montre en détail comment réaliser une représentation graphique simple des fonctions sinus et cosinus. Au départ le graphique utilisera les réglages par défaut et la figure sera ensuite améliorée pas à pas en commentant les instructions matplotlib utilisées.

Source : Matplotlib: plotting, par Nicolas Rougier, Mike Müller, Gaël Varoquaux

la licence **CC-by** Creative Commons Attribution 3.0 United States License s'applique à la traduction partielle de cette source reprise sur cette page du wiki !

## Graphe avec les réglages par défaut

- Documentation à consulter :
  - Tutoriel pyplot
  - Documentation pyplot

Matplotlib considère un ensemble de paramètres pour la personnalisation des propriétés d'une représentation graphique. Vous pouvez cependant contrôler ces paramètres par défaut pour tout : taille de la figure et sa résolution en dpi, largeur de la ligne, la couleur et le style des axes, marques et propriétés de la grille, des textes (police, taille,...), ...

Voici donc la représentation par défaut des fonctions sinus et cosinus :

<sxh python; title : 01-plotting_with_default_settings.py> #! /usr/bin/env python # -*- coding: utf-8 -*- “”“ Matplotib : Plotting with default settings Source : http://scipy-lectures.github.io/intro/matplotlib/matplotlib.html ”“”

import matplotlib.pyplot as plt # directive d'importation standard de Matplotlib import numpy as np # directive d'importation standard de numpy

X = np.linspace(-np.pi, np.pi, 256, endpoint=True) C, S = np.cos(X), np.sin(X)

plt.plot(X, C) plt.plot(X, S)

plt.show() # vue interactive de la figure

</sxh>

X est un tableau numpy de 256 valeurs comprises entre -π et +π (inclus). C est le cosinus (256 valeurs) et S est le sinus (256 valeurs).

## Instantiation des réglages par défaut

- Documentation à consulter :
  - Customizing matplotlib

Dans le programme ci-dessous, nous avons instancié (et commenté) tous les paramètres du graphique qui influencent son apparence. Les paramètres ont été définis explicitement à leurs valeurs par défaut, mais à partir de là, vous pourrez interagir avec ces valeurs afin d'explorer leur effet (voir propriétés de la ligne et styles de ligne ci-dessous).

<sxh python; title : 02-instantiating_defaults.py> #! /usr/bin/env python # -*- coding: utf-8 -*- """ Matplotib : instantiating defaults Source : http://scipy-lectures.github.io/intro/matplotlib/matplotlib.html """ import matplotlib.pyplot as plt # directive d'importation standard de Matplotlib import numpy as np # directive d'importation standard de numpy

# Create a figure of size 8×6 inches, 80 dots per inch plt.figure(figsize=(8, 6), dpi=80)

# Create a new subplot from a grid of 1×1 plt.subplot(1, 1, 1)

X = np.linspace(-np.pi, np.pi, 256, endpoint=True) C, S = np.cos(X), np.sin(X)

# Plot cosine with a blue continuous line of width 1 (pixels) plt.plot(X, C, color="blue", linewidth=1.0, linestyle="-")

# Plot sine with a green continuous line of width 1 (pixels) plt.plot(X, S, color="green", linewidth=1.0, linestyle="-")

# Set x limits plt.xlim(-4.0, 4.0)

# Set x ticks plt.xticks(np.linspace(-4, 4, 9, endpoint=True))

# Set y limits plt.ylim(-1.0, 1.0)

# Set y ticks plt.yticks(np.linspace(-1, 1, 5, endpoint=True))

# Save figure using 72 dots per inch # savefig("exercice_2.png", dpi=72)

# Show result on screen plt.show() </sxh>

## Changer la couleur et l'épaisseur des lignes

- Documentation à consulter :
  - Controlling line properties
  - Line API

Première étape, nous voulons avoir le cosinus en bleu et le sinus en rouge, et une ligne un peu plus épaisse pour chacun d'eux. Nous allons également changer légèrement la taille du graphique pour le rendre plus étendu horizontalement. Voici les lignes modifiées :

<sxh python; title : 03-changing_colors_and_line_widths.py> … plt.figure(figsize=(10, 6), dpi=80) plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-") plt.plot(X, S, color="red", linewidth=2.5, linestyle="-") … </sxh>

# Fixer les limites

- Documentation à consulter :
  - xlim() command
  - ylim() command

Les limites par défaut de la figure sont un peu juste verticalement et on souhaite un peu d'espacement pour plus de clareté. On les définit en fonction des extrema en utilisant une constante multiplicative légèrement supérieure à l'unité :

<sxh python; title : 04-setting_limits.py> … plt.xlim(X.min() * 1.1, X.max() * 1.1) plt.ylim(C.min() * 1.1, C.max() * 1.1) … </sxh>

# Fixer les graduations

- Documentation à consulter :
  - xticks() command
  - yticks() command
  - Tick container
  - Tick locating and formatting

Les graduations ne sont pas idéales car elles ne montrent pas les valeurs intéressantes (+/-π,+/-π/2) pour le sinus et le cosinus. On peut modifier les graduations de manière à ne voir que ces valeurs intéressantes :

<sxh python; title : 05-setting_ticks.py> … plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi]) plt.yticks([-1, 0, +1]) … </sxh>

From:
https://dvillers.umons.ac.be/wiki/ - **Didier Villers, UMONS - wiki**

Permanent link:
**https://dvillers.umons.ac.be/wiki/teaching:progappchim:plot_sinus_cosinus?rev=1427202691**

Last update: **2015/03/24 14:11**