Tracé de base Matplotlib

Maîtriser Matplotlib : Un guide complet pour la visualisation de données · https://moncoachdata.com/tutos/trace-de-base-matplotlib/

Le **tracé de base avec Matplotlib** repose sur un module : `matplotlib.pyplot`, qu'on importe sous l'alias `plt`. Il donne accès à toutes les fonctions de traçage. Voyons comment créer un graphique linéaire, le personnaliser et l'enregistrer.

Créer un graphique linéaire simple

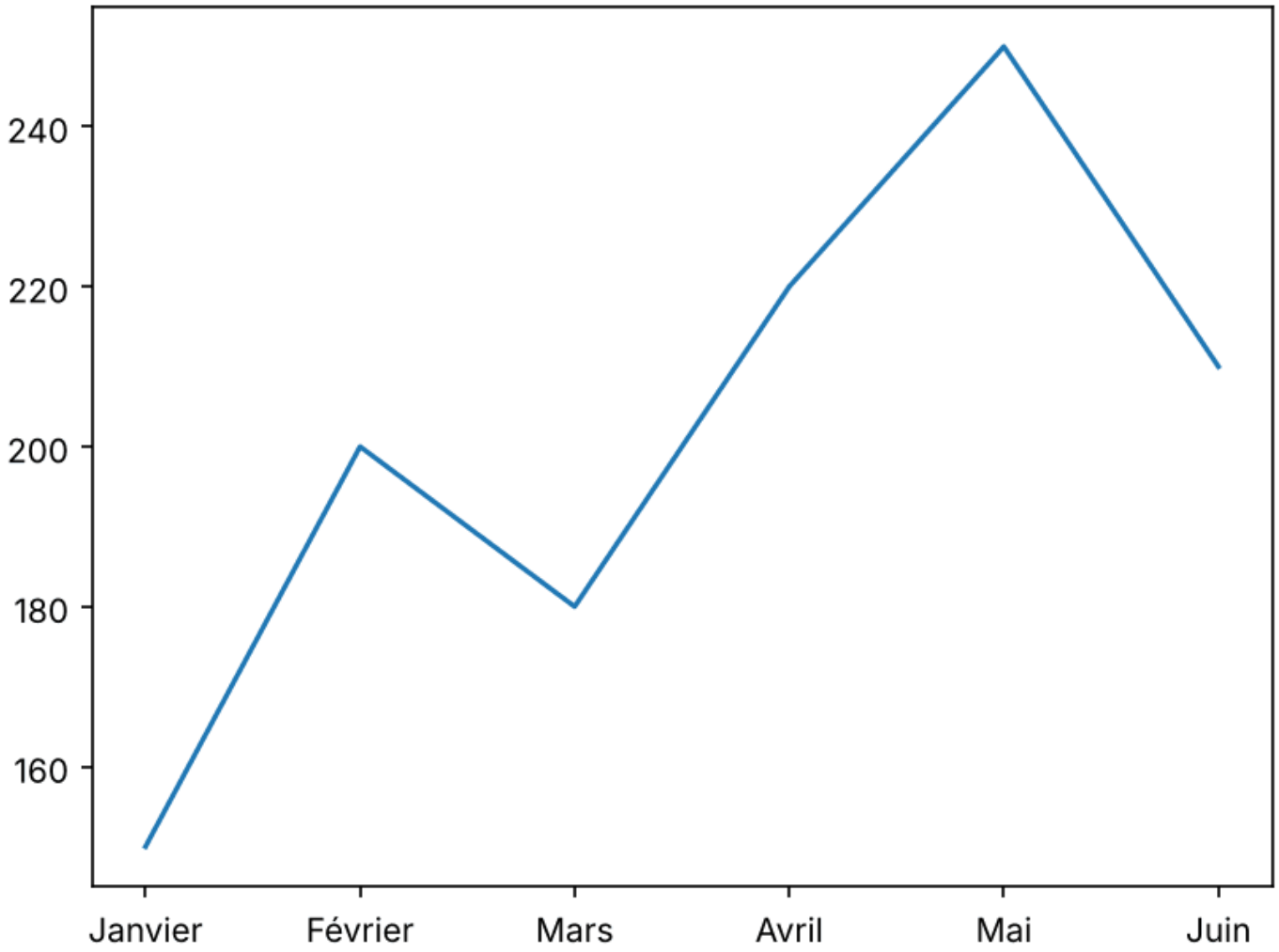
Dans un graphique linéaire, les points de données sont reliés par des lignes droites. C'est utile pour afficher des données qui varient continuellement, et repérer des tendances. Utilise `plt.plot(x, y)` pour tracer la ligne, et `plt.show()` pour afficher le graphique. Voici le trafic mensuel d'un site web (en milliers de visites) :

```python
import matplotlib.pyplot as plt

# Trafic mensuel d'un site web (en milliers)
months = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin']
traffic = [150, 200, 180, 220, 250, 210]

# Créer un graphique linéaire
plt.plot(months, traffic)
plt.show()
```

Python



Le graphique est lisible, mais il n'a ni étiquettes ni titre. Corrigeons cela.

Personnaliser les étiquettes, les couleurs et les styles

Personnaliser un tracé permet de transmettre l'information beaucoup plus clairement. Matplotlib offre de nombreuses options : couleur, style de ligne, marqueurs, et plus encore.

Ajouter des étiquettes et un titre

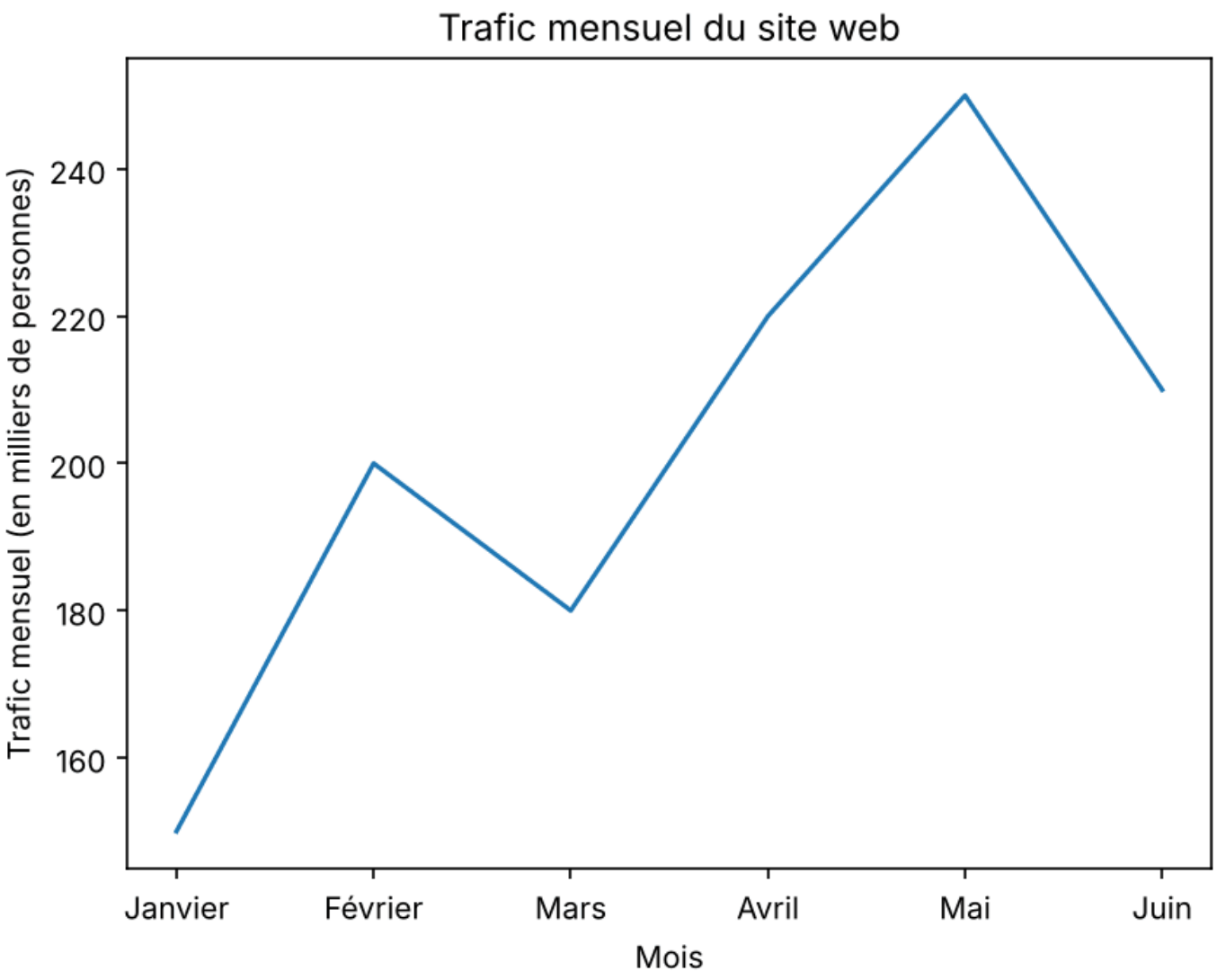
Utilise `plt.xlabel()`, `plt.ylabel()` et `plt.title()` :

```
plt.plot(months, traffic)

# Ajouter des étiquettes et un titre
plt.xlabel('Mois')
plt.ylabel('Trafic mensuel (en milliers de personnes)')
plt.title('Trafic mensuel du site web')

plt.show()
```

```
Python
```



À retenir : `plt.show()` se place à la fin de tes commandes de tracé. Dans un script, les étiquettes et les titres donnés après `plt.show()` ne s'affichent pas.

Modifier la couleur, les marqueurs et le style de ligne

Utilise le paramètre `marker` pour marquer les points, `linestyle` pour le style de ligne et `color` pour la couleur. Tu peux aussi ajouter une grille avec `plt.grid(True)`.

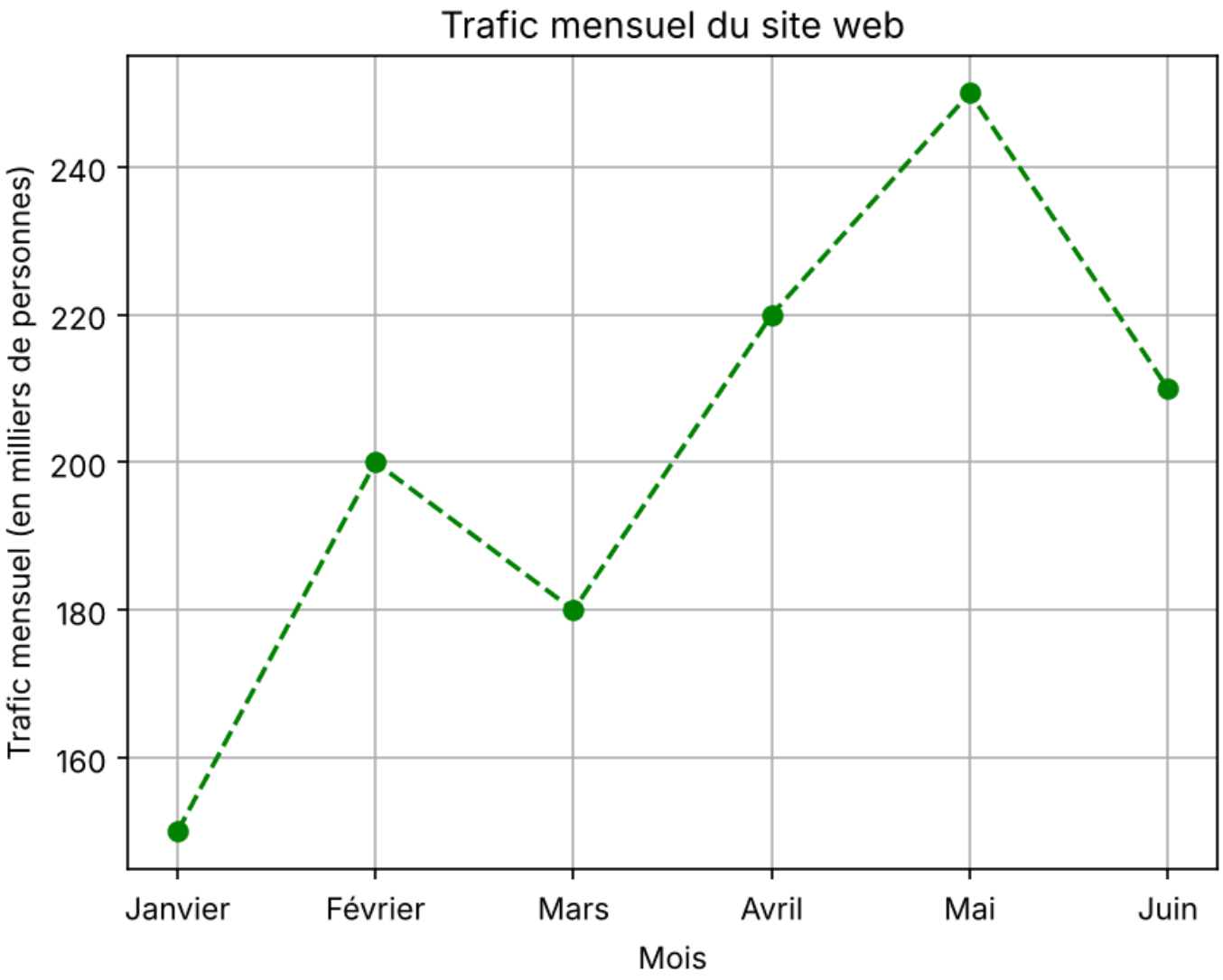
```
```

```python
# Graphique avec une apparence personnalisée
plt.plot(months, traffic, marker='o', linestyle='--', color='g')

plt.xlabel('Mois')
plt.ylabel('Trafic mensuel (en milliers de personnes)')
plt.title('Trafic mensuel du site web')

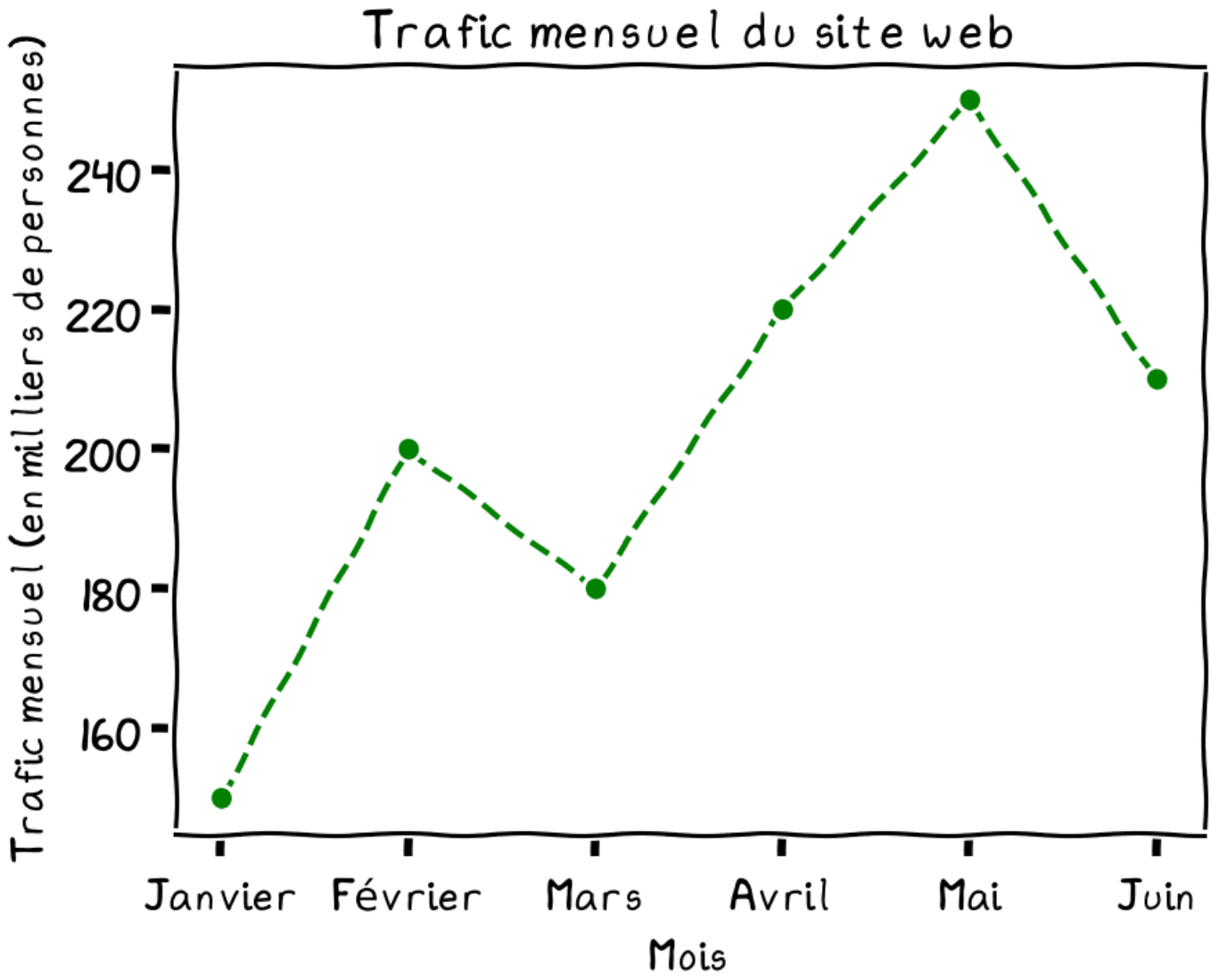
# Ajouter une grille
plt.grid(True)
plt.show()
```

Python



Changer le style du tracé

Matplotlib propose plusieurs styles prédéfinis. Pour voir ceux qui sont disponibles, utilise `plt.style.available`, puis `plt.style.use('nom_du_style')` pour appliquer un style à l'ensemble du tracé. Pour un rendu façon bande dessinée, tu peux utiliser `plt.xkcd()`. Pour revenir au style par défaut, utilise `plt.style.use('default')`. Applique toujours ces commandes de style **avant** de tracer.



Modifier la taille de la figure

Pour ajuster la taille du graphique, utilise `plt.figure(figsize=(largeur, hauteur))` (en pouces), avant l'appel à `plt.plot()`.

```python
plt.figure(figsize=(10, 5))
plt.plot(months, traffic)
plt.show()
```

Python

Tracer plusieurs lignes avec une légende

Pour tracer plusieurs lignes dans le même graphique, appelle `plt.plot()` une fois par variable. Pour les différencier, donne à chaque ligne un paramètre `label`, puis appelle `plt.legend()` :

```python
# Revenus mensuels de deux produits (en milliers d'euros)
months = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin']
product_a_revenue = [45, 55, 60, 70, 80, 75]
product_b_revenue = [35, 40, 50, 55, 70, 68]

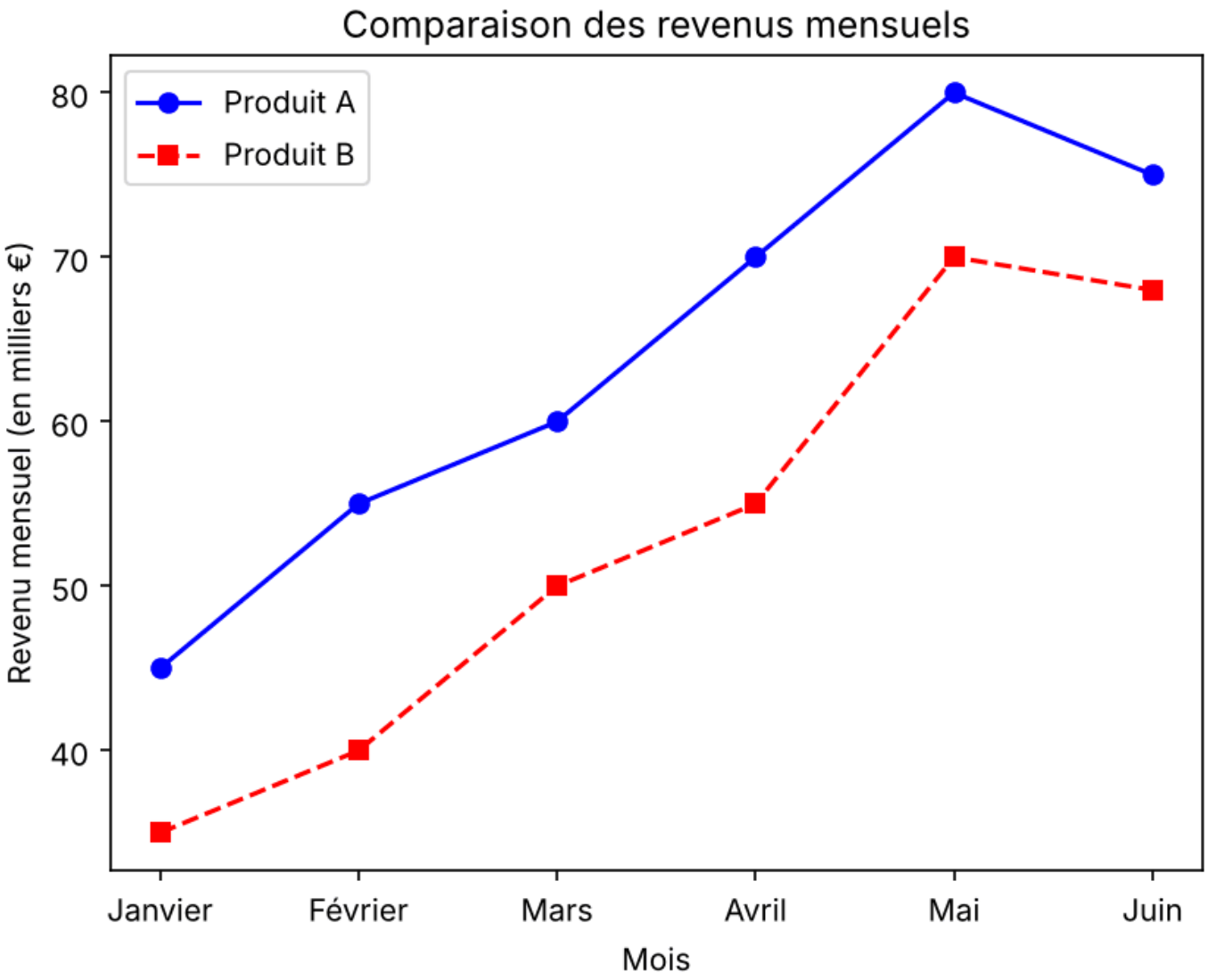
# Produit A : ligne bleue continue et marqueurs circulaires
plt.plot(months, product_a_revenue, marker='o', linestyle='-', color='blue', label='Produit A')

# Produit B : ligne rouge en pointillés et marqueurs carrés
plt.plot(months, product_b_revenue, marker='s', linestyle='--', color='red', label='Produit B')

plt.xlabel('Mois')
plt.ylabel('Revenu mensuel (en milliers €)')
plt.title('Comparaison des revenus mensuels')

# Afficher la légende pour distinguer les deux produits
plt.legend()
plt.show()
```

Python



Enregistrer un graphique Matplotlib en image

Matplotlib permet d'enregistrer un tracé sous forme de fichier image avec `plt.savefig()`. Appelle-le **avant** `plt.show()` : après, la figure est vide dans un script. Tu peux préciser le chemin, le nom et le format (.png, .jpg, .pdf, .svg).

- **Dans Jupyter Notebook :** `plt.savefig('chemin/mon_graphique.png')` ;
- **Dans Google Colab :** monte d'abord ton Google Drive, puis enregistre dedans.

```python
# Dans Jupyter Notebook
plt.savefig('mon_graphique.png', dpi=300, bbox_inches='tight')
```

Python

```python
# Dans Google Colab : monter Google Drive d'abord
from google.colab import drive
drive.mount('/content/drive')

# Enregistrer le tracé dans ton Drive (change le dossier si besoin)
plt.savefig('/content/drive/My Drive/mon_graphique.png')
```

Python

La suite de la série porte sur les types de diagrammes Matplotlib. Documentation : tutoriel pyplot.

Pour aller plus loin, consulte aussi notre guide des 9 bibliothèques Python pour la data science.

