

NOTICE D'UTILISATION DE MATPLOTLIB

D'après : <http://www.python-simple.com/python-matplotlib/scatterplot.php>

Pour réaliser des dessins ou des graphiques sous python, on peut utiliser le module « **pyplot** » de la bibliothèque « **matplotlib** »

Matplotlib est une bibliothèque python très utilisées pour réaliser des graphiques.

Le module **pyplot** regroupe un grand nombre de fonctions qui permettent de créer et personnaliser des graphiques.

Il faut tout d'abord importer le module « **pyplot** » de la bibliothèque « **matplotlib** ». Il suffit d'écrire la ligne de code suivant au début du programme.

```
import matplotlib.pyplot as plt
```

1. La première commande utile est la commande « show » qui permet d'afficher un graphique dans une fenêtre. Par défaut, celui-ci est vide. Nous devons utiliser d'autres commandes pour définir ce que nous voulons afficher.
2. La deuxième commande, est la commande « close » qui sert tout simplement à fermer la fenêtre qui s'est ouverte avec « show ». Cette commande n'est pas nécessaire car la croix en haut à droite de la fenêtre permet aussi de fermer la fenêtre ouverte. La fonction « show » est bloquante, tant que la fenêtre n'a pas été fermée, le reste du code ne s'exécute pas.
3. La troisième commande que nous allons voir est la commande « **plot** », qui permet de placer des points sur le graphique.

Voici donc un premier code :

```
import matplotlib.pyplot as plt

points = [[0.5, 1.5, "r"], [0.75, 1, "r"], [0.75, 1.75, "r"], [0.75, 2.25, "r"], [0.8, 1.4, "r"],
[0.8, 2.75, "r"], [1, 1.6, "r"], [1, 2.25, "r"], [1.25, 1.25, "r"], [1.75, 1.25, "b"], [2, 1.25,
"b"], [2, 1.5, "b"], [2.25, 1, "b"], [2.25, 1.25, "b"], [2, 1.75, "b"], [2.5, 0.5, "b"], [2.5,
1.5, "b"], [2.5, 2, "b"], [2.75, 1, "b"], [2.75, 1.75, "b"]]

point_vert = (1.5, 1.5, "g")

for u in points:
    plt.plot(u[0], u[1], u[2]+"o")

plt.plot(point_vert[0], point_vert[1], point_vert[2]+"o")

plt.show()

plt.close()
```

Copier ce code et exécutez-le.

4. On peut choisir la forme des points en utilisant par exemple l'argument « **marker** = 'D' » pour obtenir des losanges.

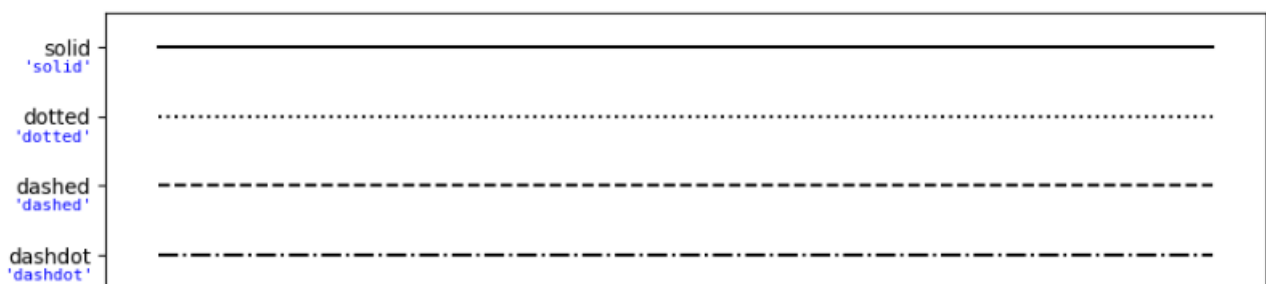
- 'o' : rond.
- 's' : carré (square).
- '+' : croix en forme de +.
- 'x' : croix en forme de x.
- '*' : étoile.
- 'D' : losange (diamond).
- 'H' : hexagone ('h' est aussi un hexagone, mais tourné).
- 'p' : pentagone.
- '.' : point.

Modifiez votre programme pour que les points soient des losanges puis des carrés.

5. On peut aussi tracer une ligne en utilisant la commande « **plt.plot** »

Ajouter la ligne de commande suivante dans votre code.

```
plt.plot([0, 5], [0, 10], color = 'red', linestyle = 'solid')
```



Essayez les différents types de tracés proposés.

6. Comment ajouter un titre au graphique ?

Il suffit d'utiliser la commande « title »

Soit : `plt.title('premiers pas avec matplotlib')`

Ajoutez cette ligne de code pour faire afficher un titre au graphique.

7. Afficher le quadrillage en utilisant la commande « **grid** ».

Soit : `plt.grid()`

Ajoutez cette ligne de code pour faire afficher le quadrillage.

8. Mettre une légende sur les axes avec la commande « xlabel » ou « ylabel ».

Soit les lignes de code suivantes :

```
plt.xlabel('abscisses')
```

```
plt.ylabel('ordonnées')
```

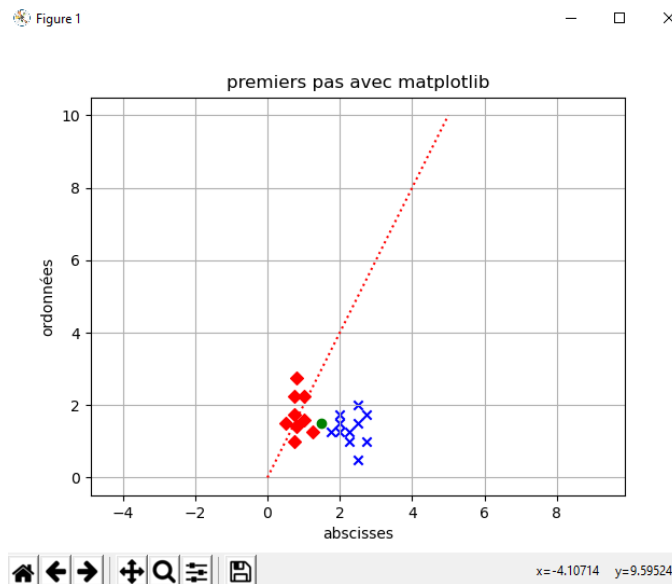
Ajoutez ces lignes de codes dans votre programme.

9. Pour obtenir un repère orthonormé soit la même échelle sur les deux axes, on utilise la commande « axis » avec le paramètre « equal ».

Soit : `plt.axis('equal')`

Ajoutez cette dernière ligne de code à votre programme et restez-le.

Voici ce que vous devez obtenir à la fin de cette activité :



Il existe bien entendu d'autres commandes pour utiliser matplotlib, et c'est en fonction des besoins que nous les découvrirons.

<https://zestedesavoir.com/tutoriels/469/introduction-aux-graphiques-en-python-avec-matplotlib-pyplot/>