

Pour ces activités tu peux utiliser le site <https://trinket.io/>

Gérer des tableaux avec Numpy

Numpy permet de manipuler plus facilement des tableaux :

Saisis ces lignes de code :

```
import numpy as np
maliste=np.linspace(0,1,num=11)
print(maliste)
```

Cela affiche le tableau maliste

```
array([0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
```

Explication :

Les deux premiers arguments sont le premier et dernier nombre de la liste. Pour un écart de 0.1 entre les éléments, il faut donc un tableau de 11 éléments.

Pour avoir 10 éléments, il faut exclure le dernier nombre de la liste :

```
maliste=np.linspace(0,1,10,endpoint=False)
```

Calcul de la somme, du produit ou la moyenne des éléments de la liste :

```
import numpy as np
maliste=np.linspace(1,10,10)
print(maliste.sum())
print(maliste.prod())
print(maliste.mean())
```

Exemple d'utilisation d'un tableau pour obtenir les différentes valeurs d'une fonction sur un intervalle :

```
import numpy as np
# définition de ma fonction
def f(x) :
    return x/2
# générer 5 valeurs allant de 10 à 20
maliste=np.linspace(10,20,5)
# obtenir les différentes valeurs de la fonction pour ce tableau
print(f(x))
```

A faire :

Soit la fonction $f(x) = \frac{x}{x^2+1}$

Ecrire le programme Python qui donne le tableau de valeur de f sur l'intervalle [a,b] avec un pas de p.
L'utilisateur doit saisir a,b et p.

Gérer des graphiques en 2D avec Matplotlib

Tracer un segment de droite de coordonnées (0,1) et (10,5)

```
import matplotlib.pyplot as plt
# segment de coordonnées (0,1) et (10,5) :
plt.plot([0,10], [1,5])
plt.grid()
# affichage du segment
plt.show()
```

Tracer un nuage de points aux coordonnées (2,3) (4,4) et (8,6) pour un repère orthonormé :

```
plt.plot([2,4,8], [3,4,6], 'ro')
plt.axis('equal')
```

Tracer des courbes qui relient des points (segments) dont les abscisses et ordonnées sont fournies dans des tableaux.

```
import numpy as np
import matplotlib.pyplot as plt
# tableau de coordonnees
x = np.array([1, 3, 4, 6])
y = np.array([2, 3, 5, 1])
plt.plot(x, y)
# affichage du segment
plt.show()
```

Tracer la fonction cosinus

```
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C = np.cos(X)
plt.plot(X, C)
```

Soit la suite définie par $u_n = \frac{20}{n^2 - 14n + 51}$

Représentez graphiquement les termes de la suite. Cette suite u est-elle croissante ?

Représentation de la fonction $f(x) = x^2$

```
x = np.linspace(-5, 5, 30)
y = x**2
plt.plot(x, y)
plt.grid()
plt.title("Titre du graphique")
plt.xlabel("abscisses")
plt.ylabel("ordonnees")
plt.show()
```

Pour afficher les axes :

```
ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))
```

A faire

Tracer la même courbe point par point sans utiliser de tableau. On utilisera une variable x qui parcourt le domaine de définition $[-5; 5]$.

A faire

Pour tout entier k compris entre -5 et 5, on note f_k la fonction définie sur l'intervalle $[0; 5]$ par $f_k(x) = k(x - 2) + 3$.

1. Tracer avec Python la représentation graphique de toutes les fonctions f_k .
2. Quelle conjecture peut-on formuler concernant ces représentations graphiques ?
3. Justifier que la conjecture faite à la question précédente est vraie.