

Dans ce TP on découvrira comment définir une fonction et tracer quelques graphiques. On aura recours aux bibliothèques **numpy** et **matplotlib**.

Définition de fonctions avec numpy :

On rappelle qu'on importe la bibliothèque **numpy** avec

```
import numpy as np
```

Pour rappel, les fonctions de références sont pour la plupart dans la bibliothèque **numpy**.

np.floor()	partir entière
np.exp()	fonction exponentielle
np.log()	fonction logarithme népérien
np.sqrt()	fonction racine carrée
abs()	valeur absolue

Exercice 1. 1. Recopier le script suivant et l'enregistrer.

```
def fonction1(a,b,c):
    y=a+b**2+c**3
    return y
```

2. Exécuter le script précédent
3. Dans la console taper fonction1(1,2,3)

À retenir

La structure de déclaration d'une fonction est :

```
def ma_fonction(var1,var2,...):
    Instruction 1
    Instruction 2
    return res1,res2,...
```

Exercice 2. 1. Créer des scripts qui permettent de calculer les images par les fonctions définies ci dessous

- ★ $f : x \mapsto e^x - \log(x^2 + 1)$
- ★ $g : x \mapsto \log|x - 1|$
- ★ $h : x \mapsto \frac{|x + 1|}{\sqrt{e^x + 1}}$

2. Calculer quelques images.

Exercice 3. Définir une fonction permettant de calculer les images par la fonction f à deux variables définie par $f : (x, y) \mapsto x^{\frac{1}{3}}y^{\frac{2}{3}}$ puis calculer quelques images.

Exercice 4. Définir une fonction qui calcule la différence entre le maximum et le minimum d'une liste de nombres et la tester avec quelques listes.

Exercice 5. Définir une fonction qui permet de calculer les images par la fonction f définie sur \mathbb{R} par $f : x \mapsto \sum_{k=0}^{10} k^x$.

Représentations graphiques avec matplotlib :

Exercice 6.

1. Créer le script :

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-3,3)
def f(x):
    y=x**2-1
    return y
y= f(x)
plt.plot(x,y)
plt.show()
```

2. Exécuter le script.
3. Modifier le pas dans la commande `arange()` pour que la courbe soit plus précise.
4. Modifier le script pour que x prenne 100 valeurs avec la commande `linspace()`.
5. Compléter les paramètres de la commande `plt.plot` pour améliorer le graphique.
Par exemple :
`plt.plot(x, y, color = 'green', linestyle = 'dashed', linewidth = 2, marker = 'o', markerfacecolor = 'blue', markersize = 5)`
6. Changer la taille du graphique avec `plt.xlim(),` et `plt.ylim(),`
7. Ajouter un titre avec `pyplot.title('...')`

À retenir

Le module `matplotlib.pyplot` s'importe avec :

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

Pour tracer une fonction d'une variable $f : [a, b] \mapsto R$ avec `matplotlib`, il faudra :

1. Créer une liste de valeurs prise par la variable x prises dans $[a, b]$ par
`x=np.arange(a, b, p)`
qui permet d'obtenir une série de valeurs dans $[a, b]$ avec un pas p . Ou bien par
`x=np.linspace(a, b, n)`
qui permet d'obtenir une série de n valeurs prises dans $[a, b]$.
2. Tracer la courbe représentative C_f de f en utilisant
`plt.plot(x, f(x))`
`plt.show()`

Exercice 7. Il est possible de tracer plusieurs courbes sur le même graphique :

1. Créer le script :

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-4,4,1000)
y1= np.exp(x)
y2= np.log(x)
y3= x
plt.plot(x,y1, 'b')
plt.plot(x,y2, 'g')
plt.plot(x,y3, 'r')
plt.show()
```

2. Créer un script qui permet de représenter les fonctions partie entière et identité sur un même graphique .

À retenir

- ★ On complète les paramètres de la commande `plt.plot` pour améliorer le graphique avec `plt.plot(x,y,color='...',linestyle='...',linewidth = ...,marker='...',markerfacecolor='...', markersize = ...)`
- ★ On change la taille du graphique avec `plt.xlim(,)` et `plt.ylim(,)`.
- ★ On ajoute un titre avec `pyplot.title("")`

Exercice 8. On considère les fonctions définies sur \mathbb{R} , par $f(x) = \frac{e^x - e^{-x}}{2}$ et $g(x) = \ln(x + \sqrt{x^2 + 1})$.

1. Étudier les variations de f .
2. Démontrer que g est la bijection réciproque de f .
3. Tracer sur la même fenêtre graphique les courbes représentatives de f et g sur l'intervalle $[-3, 3]$.
4. Ajouter un titre et une légende au tracé obtenu.

Pour plus de paramètres sur les graphiques voir :

<http://www.python-simple.com/python-matplotlib/pyplot.php>