Python pour les TP

Un exemple de programme :

```
import numpy as np
import matplotlib.pyplot as plt
plt.close('all')

U = np.array([.976, 2.99, 4.968, 7.032, 9.026]) # V
I = np.array([9.74, 29.9, 49.76, 70.19, 90.19])*1e-3 # A

R = U/I # ohm
Rexp = np.mean(R)
u_R = np.std(R,ddof=1)/np.sqrt(len(R))

print('Rexp =',Rexp,'ohm')
print('u_R =',u_R,'ohm')

plt.figure()
plt.plot(I, U, 'ob')
plt.plot(I, Rexp*I, '-r')
plt.xlabel('I (A)')
plt.ylabel('U (V)')
plt.show()
```

import des bibliothèques avec alias

fermer toutes les figures (pour ne pas se retrouver envahi)

unité en commentaire

fonction qui s'applique à une liste : parenthèse + crochet

s'applique à tous les éléments du tableau (pas comme une liste !)

tableau contenant autant d'éléments que U et I (calcul terme à terme)

nombre d'éléments de R

affichage dans la console : guillemets pour le texte directement pour les variables

créer une nouvelle figure

des ronds bleus non reliés

(abscisse, ordonnée, style)

légende des axes

une ligne rouge sans points

afficher la figure

À propos de l'écriture de code :

- Aérer le code en sautant des lignes et en mettant des espaces.
- Ne pas négliger les commentaires pour la lisbilité, en particulier pour indiquer les unités des grandeurs.
- Ne pas multiplier les parenthèses inutiles ... mais se rappeler que `x/y*z` renvoie $(xz)/y$ et non pas $x/(yz)$!
- Toujours définir une variable par grandeur, et ne jamais travailler directement avec les valeurs numériques.

À propos de numpy :

- Un tableau de valeurs numériques se construit en appliquant la fonction `np.array` à une liste : attention à n'oublier ni les parenthèses de la fonction, ni les crochets de la liste.
- Contrairement aux listes, les calculs sur les tableaux `np.array` se font toujours élément par élément. Appliquer une fonction à un tableau revient à l'appliquer à chaque élément du tableau.
- Fonctions usuelles mais pas très intuitives : `**` (puissance, p.ex. `x**2`), `np.sqrt` (racine), `np.log` (logarithme népérien), `np.log10` (logarithme décimal), `np.abs` (valeur absolue).

À propos de matplotlib.pyplot :

- Une figure doit toujours être créée (`plt.figure`) avant d'indiquer ce qui doit y apparaître. Une fois la figure complète, il faut l'afficher (`plt.show`).
- Le troisième argument de `plt.plot` renseigne sur le style de la courbe :
  → marqueur des points : `o`, `+`, `*`, etc. (rien = pas de point)
  → ligne : `-` (trait plein), `--` (pointillé), etc. (rien = pas de ligne)
  → couleur : `r` (rouge), `b` (bleu), `g` (vert), `k` (noir), etc.