



FO 6 | Outils

Python pour la physique-chimie (les bases)



I Calculs, affichage, fonctions, listes et boucles

fonctions mathématiques	<code>import math</code>	<code>math.sqrt, math.exp, math.log,</code> <code>math.pi, math.cos(radians)...</code>
une variable par grandeur, l'unité en commentaire après le symbole #	<code>m = 200e-3</code> # kg <code>v = 1.5</code> # m/s <code>Ec = 0.5 * m * v**2</code> # J	<code>200e-3 = 200 × 10⁻³;</code> séparateur décimal : le point puissance : ** (jamais <code>^</code>) : <code>.3g</code> : 3 chiffres significatifs : <code>.2f</code> : 2 chiffres après la virgule : <code>.2e</code> : écriture scientifique
f-string : f"...", variables entre accolades { }	<code>print(f"Ec = {Ec:.3g} J")</code>	: <code>.2f</code> : 2 chiffres après la virgule : <code>.2e</code> : écriture scientifique
définir une fonction : def, deux points, corps décalé de 4 espaces appel de la fonction	<code>def periode(1):</code> <code>g = 9.81</code> # m/s ² <code>return 2 * math.pi * math.sqrt(1 / g)</code> <code>T1 = periode(1.0)</code>	ne pas oublier les deux points <code>return</code> renvoie le résultat au programme (≠ <code>print</code>) définir n'affiche rien : il faut appeler la fonction
liste : valeurs entre crochets	<code>longueurs = [0.25, 0.50, 1.00]</code> # m <code>print(longueurs[0], len(longueurs))</code>	[0] : premier élément, [-1] : dernier <code>len</code> : nombre d'éléments
liste vide, à remplir	<code>periodes = []</code>	
boucle : l prend successivement chaque valeur de la liste; les lignes décalées sont répétées	<code>for l in longueurs:</code> <code>T = periode(l)</code> <code>periodes.append(T)</code> <code>print(f"l = {l:.2f} m -> T = {T:.2f} s")</code>	ajoute T à la fin de la liste
non décalée : exécutée une fois, après la boucle	<code>print("fin de la boucle")</code>	

À retenir aussi : `range(n)` fournit les n entiers $0, 1, \dots, n-1$ (utile pour répéter n fois : `for i in range(n):`).

À propos des calculs :

- Les opérations de même priorité s'effectuent de gauche à droite : $10 / 2 * 5$ vaut 25. Pour calculer $\frac{10}{2 \times 5}$, écrire $10 / (2 * 5)$. En cas de doute, **mettre des parenthèses**.
- Le signe `=` n'est pas une égalité : il *range* la valeur de droite dans la variable de gauche (`n = n + 1` a donc un sens).





II Tracer un graphique et opération sur les listes

```

import des bibliothèques [ import numpy as np
avec un alias (np, plt)   [ import matplotlib.pyplot as plt

mesures : de simples listes [ t_mes = [0.0, 0.2, 0.4, 0.6, 0.8]      # s
                             [ x_mes = [0.00, 0.21, 0.77, 1.78, 3.12]    # m

                             g = 9.81                                # m/s^2
                             t = np.linspace(0, 0.8, 100)            # s ← 100 valeurs régulièrement
                                                                           espacées de 0 à 0,8 (fin incluse)

calcul sur toutes les valeurs à la [ x = 0.5 * g * t**2                # m
fois

créer une nouvelle figure [ plt.figure()
                           [ plt.plot(t_mes, x_mes, 'o', label="mesures") ← 'o' : points non reliés
                           [ plt.plot(t, x, 'r-', label="modèle") ← 'r-' : trait plein rouge

axes toujours légendés, [ plt.xlabel("t (s)")
avec les unités          [ plt.ylabel("x (m)")
                           [ plt.legend() ← affiche les label
                           [ plt.grid() ← quadrillage : aide à la lecture

afficher la figure [ plt.show()

```

Style de tracé (3^e argument de `plt.plot`) : marqueur o, +, *... ; ligne - (pleine), -- (pointillée) ; couleur r, b, g, k (noir)...

III Lire un message d'erreur

La dernière ligne donne le type d'erreur, une flèche indique la ligne fautive.

SyntaxError	faute de « grammaire » : parenthèse non fermée, deux points oubliés...
NameError	variable inconnue : faute de frappe, ou cellule de définition pas encore exécutée
TypeError	opération impossible : v^2 au lieu de $v**2$, 2,70e3 (virgule) au lieu de 2.70e3...
IndentationError	décalage en début de ligne incorrect (boucle, fonction)
ZeroDivisionError	division par zéro

IV À propos du notebook

- Maj + Entrée exécute une cellule et passe à la suivante. Le nombre [3] indique l'**ordre** d'exécution : une cellule ne connaît que ce qui a été exécuté **avant** elle.
- En cas de comportement étrange : *Kernel* → *Restart & Run All*. Regrouper tous les `import` dans la première cellule.
- Seul le résultat de la **dernière ligne** d'une cellule s'affiche automatiquement ; sinon, utiliser `print`.