

JOURNÉES DE L'INSPECTION – RÉFORME DU LYCÉE 2019

EXERCICES PYTHON



MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE





Mise en application n°1

Les bases



Exercice 1

Objectif : S'approprier les fonctionnalités de bases en Python

1. Afficher « Bienvenue »
2. Demander à l'utilisateur de saisir une longueur d'onde en nm (mise dans une variable **longueurOnde** de type réelle)
3. Calculer la valeur de la fréquence correspondante:
 $frequence = 3.00 \times 10^8 / (longueurOnde \times 10^{-9})$ ←
4. Afficher:
« La fréquence de l'onde dont la longueur d'onde est égale à » + **longueurOnde** + « nm vaut » + **frequence** + « Hz. »

Attention:

- La “virgule” se tape: . (point)
- Multiplier se tape: *

*la fréquence devra être affichée en notation scientifique avec 3 chiffres significatifs
(voir livret page 9)*

Aide: - Livret page 8 et 9

- 1×10^9 se tape 1e9 en Python (ou 109)**

- pour transformer un nombre en texte, utiliser: str(nombre)



Exercice 2

Objectif : Traduire en langage Python l'algorithme ci-dessous permettant d'afficher la structure électronique

1. Définir la variable **numeroAt** = 20
2. Tant que **numeroAt** < 0 ou **numeroAt** > 18
Demander à l'utilisateur un entier et le mettre dans **numeroAt**

Afficher 'La configuration électronique de l'élément de numéro atomique' + **numeroAt** + 'est : '
si **numeroAt** <= 2 alors:

afficher '1s' numeroAt

si 2 < **numeroAt** <= 4 alors:

afficher '1s2 2s' + numeroAt-2

si 4 < **numeroAt** <= 10 alors:

afficher '1s2 2s2 2p' + numeroAt-4

si 10 < **numeroAt** <= 12 alors:

afficher '1s2 2s2 2p6 3s' + numeroAt-10

si 12 < **numeroAt** <= 18 alors:

afficher '1s2 2s2 2p6 3s2 3p' + numeroAt-12

**Aide pour les
conditions:
Livret page 6**



Suite de l'exercice 2 pour ceux qui veulent aller plus loin

Objectif : poursuivre le programme précédent en affichant également le symbole de l'élément chimique et son nom.

1. Créer une liste des 18 premiers éléments sous la forme de texte ("H", "He", etc...)
2. Créer une liste avec le nom des ces éléments (c'est un peu fastidieux...)
3. Créer un affichage du type
'L'élément dont le numéro atomique est Z=' (mettre le numéro atomique) 's'appelle'
(mettre le nom) ' et a pour symbole ' (mettre le symbole)
' Sa configuration électronique est : '

**Aide pour les listes:
Livret page 6 et 7**



Exercice 3 (pour les très rapides)

Objectif : Ecrire un programme calculant les angles pour la réfraction

1. Demander l'angle d'incidence en degré, l'indice du milieu 1 et celui du milieu 2
2. Transformer l'angle en radian
3. Calculer l'angle de réflexion et l'angle de réfraction
4. Les exprimer en degré et afficher un texte clair.

Attention, il faudra gérer le cas de la réflexion totale...

Aide:

- **Taper:** `import numpy as np` en début de programme
- **Le sinus de x se tape:** `np.sin(x)` (x en radians)
- **Π se tape:** `np.pi`
- **Arcsinus se tape:** `np.arcsin(valeur)`



Mise en application n°2

Numpy et Matplotlib



Exercice 4 (Numpy)

Objectif : Entrer des valeurs dans un tableau NUMPY et les manipuler

1. Importer le module NUMPY
2. Créer un tableau de date nommé t avec les valeurs suivantes : 1,2,3,4,5 jusqu'à 10
3. Créer un second tableau nommé x valant 2 fois t
4. Créer un 3^{ème} tableau nommé y valant $(-1/2)$ de $t^2 + 4$
5. Afficher les 3 tableaux
6. Afficher le dernier élément du 2nd tableau
7. Afficher le 1^{er} élément du 1^{er} tableau

Aide: page 10 du Livret

Pour aller plus loin...

1. Créer un tableau t allant de 0 à 2 par pas de 0,04
 2. Créer 2 listes contenant les équations horaires suivantes:
- $$\begin{cases} x(t) = 2t \\ y(t) = -\frac{1}{2}gt^2 + 2t + 1 \end{cases}$$

Aide: page 10 du Livret et la fonction ARANGE de Numpy



Exercice 5 (Matplotlib)

Objectif : Entrer des valeurs dans un tableau NUMPY et les manipuler

1. Reprendre le code que vous avez tapé pour l'exercice Numpy
2. Importer le module MATPLOTLIB en début de programme
3. Après la création des tableaux numpy, tracer y en fonction de x, en croix rouges.
4. Donner un nom au graphique
5. Donner des noms aux axes
6. Afficher la grille
7. Afficher la fenêtre Matplotlib

Aide: page 11 et 12 du Livret

Pour aller plus loin...

1. Reprendre le code que vous avez tapé pour l'exercice Numpy « pour aller plus loin »
2. Importer le module MATPLOTLIB en début de programme
3. Tracer les équation horaires de deux couleurs différentes sans croix mais points reliés
4. Afficher le texte « $x(t)$ » proche de la bonne courbe
5. Idem pour $y(t)$



Exercice 6 (Numpy)

Objectif : Entrer des valeurs dans un tableau NUMPY et les manipuler

1. Importer le module NUMPY
2. Créer un tableau avec les valeurs suivantes : -6.02E-1,-8.48E-1,-1.07,-1.39
3. Créer un second tableau avec les valeurs absolues du premier tableau
4. Afficher le premier et le second tableau
5. Afficher le dernier élément du 2nd tableau
6. Afficher le 1^{er} élément du 1^{er} tableau
7. Afficher tous les éléments du second tableau (en sautant une ligne entre chaque élément)

Exercice 7 (Numpy et Matplotlib)

Objectif : Afficher des données expérimentales

1. Ouvrir dans l'éditeur le fichier '**Exercice_5_matplotlib.py**'
2. Suivre les instructions en rose (avec EduPython) ou vert (avec Spyder)

Remarque: pour ne pas afficher $y(x)$ et $E_p(t)$ dans la même fenêtre, il faudra fermer la 1^{ère} pour que la seconde apparaisse

Aide: page 10, 11 et 12 du Livret



Exercice 8 (Pour les plus rapides)

Objectif : Tracer une fonction périodique dans Numpy

1. Importer le module NUMPY et MATPLOTLIB
2. Demander une période T (réel) en seconde à l'utilisateur
3. Demander une amplitude Umax en volt à l'utilisateur
4. Créer un tableau numpy appelé « t » de 200 valeurs entre 0 et 4 périodes
5. Créer une fonction U sinusoïdale: $U = U_{\max} \cdot \sin(2\pi \cdot t/T)$
6. Tracer U en fonction de t (avec grille, croix rouges, reliées ou non)

Aide: page 10, 11 et 12 du Livret

Olivier CHAUMETTE

Mathilde DIDIER-GLENAT

Jacques VINCE

Jean-Baptiste BUTET

Académie de Lyon
Version 1.3 – mai 2019

