


| | | |
|-----------------------|-----------------------------|--|
| LYCEE JOLIOT CURIE | <i>Informatique commune</i> |  |
| PTSI | DM CADEAU DE NOEL | DM |

Exercice 1 : réaliser des fiches de révision

- Dérivation numérique (explication, algorithme, dérivation par utilisation de la fonction derivative de la bibliothèque `scipy.misc`),
- Déterminer le zéro d'une fonction par la méthode de la dichotomie (explication, algorithme, utilisation de la fonction `bisect` de la bibliothèque `scipy.optimize`),
- Déterminer le zéro d'une fonction par la méthode de Newton (explication, algorithme, utilisation de la fonction `newton` de la bibliothèque `scipy.optimize`),
- Permutations (voir S1_8, algorithmes structure séquentielle par boucles imbriquées),
- Utilisation de modules et de bibliothèques
 - Récupérer les données qui se trouvent dans un fichier,
 - Ordonner les données récupérées,
 - Traiter les données,
 - Module `math`,
 - Donner 3 fonctions importantes du module `math`.
 - Module `scipy`,
 - Utilisation de la fonction `linregress()` du module `scipy.stats`,
 - Module `statistics`,
 - Donner 2 fonctions importantes du module `statistics`.
- Algorithmes opérant sur une structure séquentielle par boucles imbriquées (l'évocation de la complexité dans la fiche de révision est impérative),
- Tri à bulles (explication + algorithme, version non optimisée),
- Images.

Exercice 2 : divers

- Profiter de sa famille et de ses amis,
- Se reposer,
- Faire du sport,
- Sortir,
- Lire,
- Bien se couvrir, dehors vous allez avoir si froid. C'est un peu à cause de M HERY,
- JOYEUX NOEL !
- ...

Exercice 3 : programme mystère

Que fait le programme suivant ? (voir au dos).

```

"""
IMPORTATION DES BIBLIOTHEQUES UTILES
"""
import matplotlib.pyplot as plt
import numpy as np
"""
PARAMETRES
"""
nombre=5 #valeur mini : 3
h_element=40 #unité en pixel
"""
DEFINITION DES FONCTIONS SECONDAIRES
"""
def tableau_vierge(taille):
    #
    return np.ones((taille,taille,3))

def fig_geometrique_1(tableau,point_depart_li,point_depart_col):
    #
    ##
    for i in range(0,h_element):
        ##
        ##
        for j in range(-i//3-point_depart_li//5,\
                      +i//3+point_depart_li//5+1):
            #
            tableau[point_depart_li+i][point_depart_col+j]=0.8,0.8,0.5
    return tableau

def fig_geometrique_2(tableau,point_depart_li,point_depart_col):
    #
    ##
    for i in range(0,h_element):
        ##
        for j in range(point_depart_col-point_depart_li//20,\
                      point_depart_col+point_depart_li//20):
            #
            tableau[point_depart_li+i][j]=0.6,0.6,0.4
    return tableau
"""
DEFINITION DE LA FONCTION PRINCIPALE
"""
def figure_mystere(nombre,h_element):
    #
    taille=(nombre+1)*h_element
    #
    image=tableau_vierge(taille)
    #
    for i in range(0,taille-h_element,h_element):
        fig_geometrique_1(image,i,taille//2)
    #
    fig_geometrique_2(image,taille-h_element,taille//2)
    return image
"""
EXECUTION DE LA FONCTION PRINCIPALE ET AFFICHAGE
"""
im=figure_mystere(nombre,h_element)
plt.imshow(im)
plt.show()

```